

1. Гарантированная точность решения систем линейных уравнений/Годунов С. К., Антонов А. Г., Кирилюк О. П., Костин В. И.—Новосибирск: Наука. Сиб. отд-ние, 1988.
2. Булгаков А. Я. Эффективно вычисляемый параметр качества устойчивости систем линейных дифференциальных уравнений с постоянными коэффициентами // Сиб. мат. журн.— 1980.— Т. 21, № 3.— С. 32—41.
3. Godunov S. K., Boulgakov A. J. Difficultes calculatives dans le problemes de Hurwitz et methodes a les surmontes // Analysis and Optimization of Systems: Proc./Fifth Intern. Conf. Versailles; 1982.— Berlin a. o.: Springer, 1982.— P. 845—851.— (Lecture Notes in Control and Information Sciences; 44).
4. Годунов С. К. Задача о дихотомии спектра матрицы // Сиб. мат. журн.— 1986.— Т. 27, № 5.— С. 24—37.
5. Булгаков А. Я., Годунов С. К. Круговая дихотомия матричного спектра.—Новосибирск, 1987.— 34 с.— (Препринт/АН СССР. Сиб. отд-ние. Ин-т математики; № 5).
6. Булгаков А. Я. Оценка матрицы Грина, теоремы непрерывности матрицы Грина и параметра дихотомии.—Новосибирск, 1987.— 36 с.—(Препринт/АН СССР. Сиб. отд-ние. Ин-т математики; № 6).
7. Булгаков А. Я. Точность расчета проекторов $\Pi_+(A)$, $\Pi_-(A)$ — Новосибирск, 1987.— 49 с.— (Препринт/АН СССР. Сиб. отд-ние, Ин-т математики; № 14).
8. Малышев А. Н. Факторизация матричных полиномов // Сиб. мат. журн.— 1982.— Т. 23, № 3.— С. 136—146.

С. И. ФАДЕЕВ

ПРОГРАММА ЧИСЛЕННОГО РЕШЕНИЯ НЕЛИНЕЙНЫХ КРАЕВЫХ ЗАДАЧ ДЛЯ СИСТЕМ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ С ПАРАМЕТРОМ

В предлагаемой работе рассматривается вычислительная сторона проблемы продолжения решения нелинейной краевой задачи по скалярному параметру Q . Краевая задача формулируется для системы обыкновенных дифференциальных уравнений на конечном отрезке с двухточечными краевыми условиями. Во многих приложениях этой проблемы параметр Q входит естественным образом в формулировку краевой задачи и при этом имеет место непрерывная зависимость решения от параметра. Метод продолжения может оказаться полезным в случае, когда требуется изучить поведение решения в зависимости от значений Q . В другом случае метод продолжения применяется для нахождения достаточно хорошего начального приближения, позволяющего численно построить решение краевой задачи при заданном значении Q итерационным способом. В силу нелинейности краевой задачи и в том и другом случае при продолжении решения по параметру могут возникнуть затруднения вычислительного характера, связанные с проблемой множественности решений.

По существу, в работе излагаются метод и программная реализация численного решения системы трансцендентных уравнений с параметром Q , являющейся дискретной моделью краевой задачи. Продолжение решения по параметру в сочетании с методом Ньютона — Канторовича сопровождается процедурой параметризации. Это позволяет, решая систему трансцендентных уравнений, построить часть гладкой пространственной кривой, задаваемой системой при помощи параметра Q , между двумя особыми точками, если таковые имеют место. Таким образом, в работе рассматривается только та множественность решений, которая возникает в случае неоднозначной зависимости функций от параметра Q , описывающих гладкую пространственную кривую.

Эффективность программы связана с учетом структуры матрицы якобиана системы трансцендентных уравнений при реализации предлагаемого варианта продолжения решения. Кроме того, в программе заложена возможность адаптации сетки, на которой формулируется дискретная модель, в процессе построения пространственной кривой, а также адаптация шага по параметру продолжения решения, определяемого параметризацией.

Для автора были полезны обсуждения с В. Л. Мирошниченко выбранного способа адаптации сетки. Детали данной работы неоднократно обсуждались с С. К. Годуновым, которому автор выражает благодарность за внимание и поддержку.

§ 1. ПРЕДВАРИТЕЛЬНЫЕ ЗАМЕЧАНИЯ

В данной работе дается подробное описание программной реализации численного метода решения нелинейной краевой задачи для системы из n обыкновенных дифференциальных уравнений на конечном интервале (a, b) изменения скалярного аргумента x :

$$\frac{dy}{dx} = f(x, y, Q), \quad x \in (a, b), \quad (1)$$

$$l(y, Q) = 0, \quad \text{при } x = a, \quad g(y, Q) = 0 \quad \text{при } x = b.$$

Здесь y и f — векторы размерности n , Q — скалярный параметр, $f(x, y, Q)$, $l(y, Q)$, $g(y, Q)$ — достаточно гладкие вектор-функции по совокупности аргументов, причем $l(y, Q)$ размерности n_0 и $g(y, Q)$ размерности $n - n_0$ определяют n независимых краевых условий. Отметим, что полная априорная информация о решении (1) имеется лишь в исключительных случаях. Поэтому в рамках достаточно общей формулировки двухточечной краевой задачи мы будем исходить из типичной ситуации, когда изучение решения (1) носит характер численного эксперимента. Для конкретности будем предполагать, что (1) имеет единственное решение, если Q принадлежит достаточно малой окрестности точки Q_0 .

Как известно, мощным средством численного исследования нелинейных проблем является метод Ньютона — Канторовича. В дальнейшем мы будем рассматривать его применение для решения следующей задачи, часто встречающейся в приложениях. Пусть требуется построить численное решение (1) $y = y(x, Q)$ при $Q = \bar{Q}$. Однако данные, которыми мы располагаем, позволяют найти $y(x, Q_0)$, а приемлемое начальное приближение решения (1) при $Q = \bar{Q}$ нам неизвестно.

Имея в распоряжении программную реализацию метода Ньютона — Канторовича, дополненную процедурой продолжения решения по параметру, мы можем воспользоваться ею, чтобы попытаться численно продолжить решение (1) по параметру Q , устремив Q к \bar{Q} . Численный эксперимент окажется успешным, если при $Q \in [Q_0, \bar{Q}]$ краевая задача имеет, например, единственное решение (для определенности пусть $\bar{Q} > Q_0$). Попутно мы имеем возможность численно исследовать зависимость решения (1) от параметра Q , что представляет самостоятельный интерес.

Один из способов решения краевой задачи состоит в приближенном представлении (1) в виде конечномерной системы трансцендентных уравнений

$$\Phi(Y, Q) = 0, \quad (2)$$

где $\Phi(Y, Q)$ — достаточно гладкая вектор-функция размерности N векторного аргумента Y размерности N и скалярного аргумента Q . Затем для решения (2) применяется метод Ньютона — Канторовича: если Y^0 — приближение решения системы (2), то поправка Z , уточняющая Y^0 ,

находится из системы линейных алгебраических уравнений

$$\Phi_Y(Y^0, Q)Z = -\Phi(Y^0, Q), \det \Phi_Y(Y^0, Q) \neq 0,$$

где Φ_Y — матрица якобиана; вектор $Y^0 + Z$ объявляется исходным приближением решения (2) и так далее, пока норма поправки не окажется достаточно малой. При этом вектор $Y^0 + Z$ принимается за приближенное решение (2). Пусть при $Q \in [Q_0, \bar{Q}]$ система (2) имеет единственное решение, являющееся гладкой вектор-функцией $Y = Y(Q)$, и пусть $Y(Q_0)$ — решение (2) при $Q = Q_0$. Построение $Y(Q)$ при помощи процедуры продолжения решения (2) по параметру Q можно сформулировать как задачу Коши:

$$\begin{aligned} Y_Q &= -[\Phi_Y(Y, Q)]^{-1}\Phi_Q(Y, Q), \\ Y|_{Q=Q_0} &= Y(Q_0), \quad Q_0 \leq Q \leq \bar{Q}, \end{aligned} \quad (3)$$

где Y_Q и Φ_Q — векторы производных по Q .

Когда единственность решения (1) не гарантируется, реализация метода осложняется в случае существования точек ветвления решений по параметру Q . При стремлении Q к \bar{Q} мы можем попасть в окрестность точки ветвления Q_* , $Q_* \in [Q_0, \bar{Q}]$, где применение метода Ньютона — Канторовича становится некорректным, и, следовательно, требуются иные способы продолжения решения. Примером является случай, когда при $Q = \bar{Q}$ решение задач (1) не существует вообще (см. (5)). Заметим, что продолжение решения по параметру Q , основывающееся на методе Ньютона — Канторовича, остается эффективным методом и при множественности решений до тех пор, пока в процессе построения какой-либо ветви $y = y(x, Q)$ мы не попадаем в окрестность точки ветвления. Вычислительные затруднения здесь могут быть связаны с выбором начального приближения решения (1) для стартового значения $Q = Q_0$.

Возможности обсуждаемого метода существенно расширяются за счет параметризации — процедуры поиска параметров задачи (1), которые наряду с Q могут быть использованы для продолжения решения. Рассмотрим краевую задачу, порождаемую (1):

$$\frac{dy}{dx} = f(x, y, Q), \quad x \in (a, b), \quad c \in (a, b), \quad (4)$$

$$l(y, Q) = 0 \text{ при } x = a, \quad g(y, Q) = 0 \text{ при } x = b, \quad y_k = \lambda \text{ при } x = c.$$

Здесь λ — заданное значение k -й компоненты $y_k(x)$ вектор-функции $y(x)$ при заданном значении $x = c$, а параметр Q вместе с $y(x)$ подлежат определению. Пусть (4) имеет единственное решение при λ из некоторой окрестности точки λ_0 . Тогда $y_k(c) = \lambda$ можно выбрать в качестве параметра задачи (4) для продолжения решения: $y = y(x, \lambda)$, $Q = Q(\lambda)$. Если при этом λ_0 соответствует значению $Q = Q_0$ краевой задачи (1), то $y = y(x, \lambda)$ является продолжением решения (1) по параметру λ , где $Q = Q(\lambda)$. Пусть λ_* — точка ветвления решений по параметру λ . Применение параметризации оказывается полезным в том случае, если $Q(\lambda_*) \neq Q_*$ и продолжение решения по параметру λ после его выбора реализуется столь же просто, как и по параметру Q . Естественно, что параметризация не устраняет всех проблем, связанных с ветвлением решений (1). Имеется в виду лишь ветвление решений определенного типа, на описании которого мы сейчас остановимся.

Пусть каждая из компонент $y_j(x, Q)$ вектор-функции $y(x, Q)$, $(x, Q) \in (a, b) \times [Q_0, \bar{Q}]$, $j = 1, 2, \dots, n$, являющаяся решением (1), задает гладкую поверхность, допускающую параметрическое представление в виде однозначных функций. В сечении плоскостью $Q = \text{const}$ эта поверхность может содержать одну или несколько кривых: $y_{j1} = y_{j1}(x, Q)$, $y_{j2} = y_{j2}(x, Q)$ и т. д. Из геометрической интерпретации ясно, что в этом случае плоскость $Q = Q_*$ является касательной к поверхности $y_i =$

$= y_j(x, Q)$ в точках кривой $y_j = y_j(x, Q_*)$. Пусть в сечении $x = \text{const}$ характер касания таков, как это представлено на рис. 1. Тогда поверхности $y_j(x, Q)$ можно построить численно при помощи параметризации.

Предположим, что решениями задачи (1) являются несколько вектор-функций $y^{[1]}(x, Q)$, $y^{[2]}(x, Q)$ и т. д., и пусть компоненты каждой вектор-функции описывают совокупности гладких поверхностей с указанными свойствами. Если поверхности одной совокупности не имеют общих точек с соответствующими поверхностями другой совокупности, то для их численного построения также применима параметризация.

В качестве иллюстрации рассмотрим нелинейную краевую задачу для дифференциального уравнения второго порядка, которую запишем в виде (1):

$$\frac{dy_1}{dx} = y_2, \quad \frac{dy_2}{dx} = -\frac{Q}{(1-y_1)^2}, \quad x \in (0, 1), \quad (5)$$

$$y_2(0) = y_1(1) = 0,$$

где $Q \geq 0$, $0 \leq y_1 < 1$. Приведем точное решение (5), определяющее x и y_2 как функции y_1 и параметра λ :

$$x(y_1, \lambda) = \frac{1}{\rho} \left[\sqrt{(1-y_1)(\lambda-y_1)} + (1-\lambda) \ln \frac{\sqrt{1-y_1} + \sqrt{\lambda-y_1}}{\sqrt{1-\lambda}} \right],$$

$$y_2(y_1, \lambda) = -\rho \sqrt{\frac{1-y_1}{1-\lambda}}, \quad \rho = \sqrt{\lambda} + (1-\lambda) \ln \frac{1+\sqrt{\lambda}}{\sqrt{1-\lambda}}, \quad Q(\lambda) =$$

$$= \frac{1-\lambda}{2} \rho^2, \quad 0 < \lambda < 1, \quad 0 \leq y_1 \leq \lambda.$$

Если $y_1 = \lambda$, то $x = 0$, так что λ — задаваемое значение y_1 при $x = 0$.

Множественность решений (5) наглядно представляется графиком функции $Q(\lambda)$, приведенным на рис. 2. Эта кривая принадлежит поверхности $y_1(x, Q)$, задаваемой, как и поверхность $y_2(x, Q)$, решением краевой задачи в сечении плоскостью $x = 0$. Поверхность $y_1(x, Q)$ приведена на рис. 3. Здесь плоскость $Q = Q_*$, $Q_* = 0.35$, содержащая график решения (5) при $\lambda = 0.3883$, является касательной. Таким образом, при $Q < Q_*$ краевая задача имеет два решения, а при $Q > Q_*$ решения (5) не существует.

Заметив, что $y_1(x, Q) = y_2(x, Q) \equiv 0$ при $Q = Q_0 = 0$, рассмотрим на этом примере различные ситуации, связанные с продолжением решения. Из рис. 2 видно, что продолжение решения (5) по параметру Q возможно до некоторого значения $Q = \bar{Q} < Q_*$. Пусть \bar{Q} отвечает значение $\lambda = \bar{\lambda}$, $\bar{\lambda} = y_1(0, Q)$. Чтобы избежать затруднений вычислительного характера, связанных с ветвлением решения по параметру Q при $Q = Q_*$,

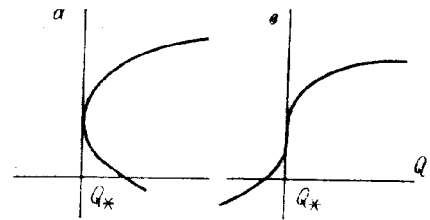


Рис. 1.

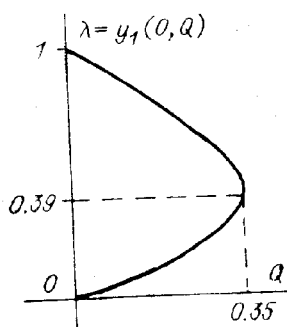


Рис. 2.

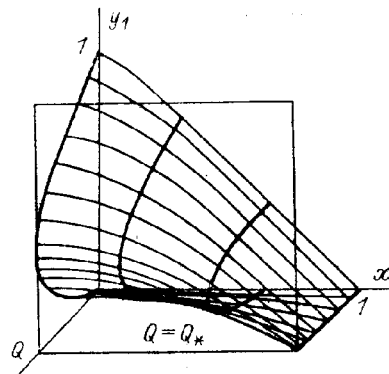


Рис. 3.

выберем λ в качестве параметра и, начиная с $\lambda = \bar{\lambda}$, продолжим решение (5) уже по параметру λ вплоть до $\lambda = 1$. Как следует из точного решения, при этом мы не встретимся с ветвлением решения по параметру λ . Более того, мы могли бы продолжить решение (5) по λ , начиная с $\lambda = 0$, которое соответствует $Q = 0$.

В целях применения параметризации преобразуем (5) к нелинейному интегральному уравнению относительно функции $u(x) \equiv y_1(x)$:

$$u(x) = Q \int_0^1 G(x, t) \frac{dt}{[1 - u(t)]^2}. \quad (6)$$

Здесь $G(x, t)$ — функция Грина краевой задачи

$$\frac{d^2 u}{dx^2} = 0, \quad x \in (0, 1), \quad \frac{du}{dx}(0) = u(1) = 0,$$

имеющая вид

$$G(x, t) = \begin{cases} 1 - t, & x \leq t, \\ 1 - x, & x > t. \end{cases}$$

Теперь чтобы получить формулировку краевой задачи (4), соответствующую (6), достаточно в (6) положить $x = 0$ и воспользоваться равенством

$$\lambda = u(0) = Q \int_0^1 G(0, t) \frac{dt}{[1 - u(t)]^2}$$

для исключения Q . В результате приходим к уравнению с параметром λ следующего вида:

$$u(x) = \lambda \frac{\int_0^1 G(x, t) \frac{dt}{[1 - u(t)]^2}}{\int_0^1 G(0, t) \frac{dt}{[1 - u(t)]^2}}. \quad (7)$$

Решению $u(x, \lambda)$ уравнения (7) отвечает значение $Q(\lambda)$, вычисляемое по формуле

$$Q(\lambda) = \frac{\lambda}{\int_0^1 G(0, t) \frac{dt}{[1 - u(t)]^2}}.$$

Аналогичное представление имеет место и в более общем случае, когда нелинейная краевая задача формулируется для дифференциального уравнения n -го порядка

$$\mathcal{P}_n \left(x, \frac{d}{dx} \right) u + Qq(x, u) = 0, \quad x \in (a, b),$$

с однородными краевыми условиями. Здесь $\mathcal{P}_n \left(x, \frac{d}{dx} \right)$ — линейный непрерывный дифференциальный оператор n -го порядка, $q(x, u)$ — достаточно гладкая функция x и u . Предполагается, что при $Q = 0$ краевая задача имеет только тривиальное решение.

Один из подходов к численному решению (7) состоит в следующем. Введем сетку по x с узлами x_j :

$$0 = x_1 < x_2 < \dots < x_m = 1, \quad h_i = x_{i+1} - x_i,$$

и приближенно заменим функцию $q(x) \equiv [1 - u(x)]^{-2}$ интерполяционным кубическим сплайном $T(x)$ дефекта 1,

$$T(x) = \sum_{k=1}^m q(u_k) \varphi_k(x), \quad u_k = u(x_k),$$

с погрешностью интерполяции порядка h_{\max}^4 , $h_{\max} = \max_i h_i$. Здесь $\varphi_k(x)$ — k -й фундаментальный сплайн, определяемый следующими условиями интерполяции:

$$\varphi_k(x_i) = \begin{cases} 1, & i = k, \\ 0, & i \neq k, \end{cases}$$

$$\frac{d\varphi_k}{dx}(x_1) = 0, \quad \frac{d^3\varphi_k}{dx^3}(x_{m-1} - 0) = \frac{d^3\varphi_k}{dx^3}(x_{m-1} + 0).$$

В левом граничном условии интерполяции учтено, что $\frac{dq}{dx}(x_1) = 0$. Правое граничное условие требует непрерывности третьей производной $\varphi_k(x)$ в точке x_{m-1} . На каждом из отрезков $[x_i, x_{i+1}]$ фундаментальный сплайн $\varphi_k(x)$ можно определить формулой

$$x_i \leq x \leq x_{i+1}, \quad x = x_i + h_i t, \quad 0 \leq t \leq 1,$$

$$\varphi_k(x) = \varphi_k(x_i)(1-t) + \varphi_k(x_{i+1})t - h_i^2 t(1-t) [(2-t)M_i^{(k)} + (1+t)M_{i+1}^{(k)}].$$

Коэффициенты $M_1^{(k)}, M_2^{(k)}, \dots, M_m^{(k)}$ удовлетворяют системе из m линейных алгебраических уравнений с трехдиагональной матрицей, порожденной условиями интерполяции и требованием непрерывности первой производной сплайна во всех внутренних точках разбиения отрезка по x (непрерывность второй производной обеспечивается самой формулой). С учетом граничных условий интерполяции имеем

$$2h_1 M_1^{(k)} + h_1 M_2^{(k)} = r_1^{(k)},$$

$$h_1 M_1^{(k)} + 2(h_1 + h_2) M_2^{(k)} + h_2 M_3^{(k)} = r_2^{(k)} - r_1^{(k)},$$

$$\dots$$

$$h_{m-4} M_{m-4}^{(k)} + 2(h_{m-4} + h_{m-3}) M_{m-3}^{(k)} + h_{m-3} M_{m-2}^{(k)} = r_{m-3}^{(k)} - r_{m-4}^{(k)},$$

$$h_{m-3} M_{m-3}^{(k)} + 2(h_{m-3} + h_{m-2}) M_{m-2}^{(k)} = r_{m-2}^{(k)} - h_{m-2} M_{m-1}^{(k)},$$

где

$$r_i^{(k)} = \frac{\varphi_k(x_{i+1}) - \varphi_k(x_i)}{h_i}, \quad M_{m-1}^{(k)} = \frac{\varphi_k(x_m) - 2\varphi_k(x_{m-1}) + \varphi_k(x_{m-2})}{6h_{m-1}}.$$

Разрешив систему относительно $M_1^{(k)}, M_2^{(k)}, \dots, M_{m-2}^{(k)}$, найдем $M_m^{(k)} = 2M_{m-1}^{(k)} - M_{m-2}^{(k)}$. Обратим внимание на то, что при формировании системы использовано необременительное ограничение на разбиение отрезка по x : $h_{m-2} = h_{m-1}$. Важно отметить, что матрица системы имеет свойство диагонального преобладания.

Итак, в узлах сетки уравнение (6) приближенно преобразуется к виду

$$\lambda = u_1 = Q \sum_{k=1}^m A_{1k} q(u_k), \quad u_m = 0,$$

$$u_i = Q \sum_{k=1}^m A_{ik} q(u_k), \quad i = 2, 3, \dots, m-1,$$

где

$$A_{jk} = \int_0^1 G(x_j, t) \varphi_k(t) dt.$$

После исключения Q получим следующую систему трансцендентных уравнений, содержащую параметр λ , относительно сеточных значений $u(x)$:

$$u_i = \lambda \frac{\sum_{k=1}^m A_{ik} q(u_k)}{\sum_{k=1}^m A_{1k} q(u_k)}, \quad i = 2, 3, \dots, m,$$

или

$$\Phi(X, \lambda) = 0, \quad (8)$$

где X — вектор с компонентами u_2, u_3, \dots, u_{m-1} . При этом параметр Q вычисляется по формуле $Q = \lambda \left| \sum_{k=1}^m A_{1k} q(u_k) \right|$.

Во многих интересных приложениях функция Грина имеет простое выражение, позволяющее точно найти представление коэффициентов A_{jk} через коэффициенты $M_i^{(k)}$ фундаментального сплайна $\varphi_k(x)$. Вообще говоря, A_{jk} определяются как сеточные значения функции $v_{(x)}^{(k)}$, являющейся решением линейной краевой задачи для уравнения

$$\mathcal{P}_n \left(x, \frac{d}{dx} \right) v^{(k)} + \varphi_k(x) = 0,$$

так что $A_{jk} = v^{(k)}(x_j)$.

К системе (8) применим метод продолжения решения по параметру λ в виде (3):

$$X_\lambda = -[\Phi_x(X, \lambda)]^{-1} \Phi_\lambda(X, \lambda), \quad X|_{\lambda=0} = 0, \quad 0 < \lambda < 1.$$

Здесь Φ_x — матрица якобиана, X_λ и Φ_λ — векторы производных по параметру λ .

В данном примере параметризация состояла в однократной замене параметра краевой задачи: вместо Q был выбран параметр λ , $\lambda = y_1|_{x=0}$. Однако в общем случае для продолжения решения (2) по параметру требуется регулярное применение процедуры параметризации. Далее будет рассмотрена дискретная модель (1) в виде системы (2) с разреженной матрицей якобиана Φ_Y и указана процедура параметризации, при которой переход к дискретной модели (4) происходит с учетом разреженности Φ_Y .

§ 2. ДИСКРЕТНАЯ МОДЕЛЬ КРАЕВОЙ ЗАДАЧИ

Представим (1) в виде системы интегральных уравнений

$$y(x) = y(x_i) + \int_{x_i}^{x_{i+1}} f[t, y(t), Q] dt, \quad x_i \leq x \leq x_{i+1},$$

где x_i — узлы сетки

$$a = x_1 < x_2 < \dots < x_m = b, \quad h_i = x_{i+1} - x_i.$$

Сеточные значения вектор-функции $y(x)$ удовлетворяют равенствам

$$\begin{aligned} l[y(x_1), Q] &= 0, \\ y(x_{i+1}) &= y(x_i) + \int_{x_i}^{x_{i+1}} f[t, y(t), Q] dt, \quad i = 1, 2, \dots, m-1. \\ g[y(x_m), Q] &= 0. \end{aligned} \quad (9)$$

Приближенно заменим здесь интеграл на квадратурную формулу. Например, по формуле трапеций приближенное выражение интеграла с погрешностью порядка h_i^3 имеет вид

$$\int_{x_i}^{x_{i+1}} f[t, y(t), Q] dt \approx \frac{h_i}{2} [f(x_i, y^i, Q) + f(x_{i+1}, y^{i+1}, Q)],$$

где $y^k = y(x_k)$. В результате получим систему уравнений относительно векторов y^1, y^2, \dots, y^m .

Рассмотрим в качестве дискретной модели (1) систему уравнений, порождаемую формулой Симпсона. С погрешностью порядка h_i^5 имеем

$$\int_{x_i}^{x_{i+1}} f[t, y(t), Q] dt \approx \frac{h_i}{6} [f(x_i, y^i, Q) + 4f(\bar{x}_i, \bar{y}^i, Q) + f(x_{i+1}, y^{i+1}, Q)],$$

где $\bar{x}_i = (x_i + x_{i+1})/2$. Для определения \bar{y}^i воспользуемся кубической параболой Эрмита, аппроксимирующей $y(x)$ на отрезке $[x_i, x_{i+1}]$ с погрешностью порядка h_i^4 :

$$x_i \leq x \leq x_{i+1}, \quad x = x_i + h_i \tau, \quad 0 \leq \tau \leq 1,$$

$$y(x) \approx (1 - \tau)y^i + \tau y^{i+1} + \tau(1 - \tau)[(1 - \tau)a^i + \tau b^i],$$

где

$$a^i = y^i - y^{i+1} + h_i f(x_i, y^i, Q), \quad a^{i+1} = y^{i+1} - y^i - h_i f(x_{i+1}, y^{i+1}, Q).$$

Полагая здесь $\tau = 1/2$, получим

$$\bar{y}^i = \frac{1}{2}(y^i + y^{i+1}) + \frac{h_i}{8}[f(x_i, y^i, Q) - f(x_{i+1}, y^{i+1}, Q)].$$

Точность формулы Симпсона при этом сохраняется. После замены в (9) интеграла на квадратурную формулу приходим к следующей системе уравнений:

$$\begin{aligned} l(y^1, Q) &= 0, \\ F^i &= y^i - y^{i+1} + \frac{h_i}{6}[f(x_i, y^i, Q) + 4f(\bar{x}_i, \bar{y}^i, Q) + \\ &+ f(x_{i+1}, y^{i+1}, Q)], \quad i = 1, 2, \dots, m-1, \\ g(y^m, Q) &= 0. \end{aligned} \quad (10)$$

Система (10) определяет n компонент векторов y^k , $k = 1, 2, \dots, m$, т. е. ее размерность N равна mn .

Пусть $y_1^k, y_2^k, \dots, y_n^k$ — компоненты вектора y^k ; $F_1^j, F_2^j, \dots, F_n^j$ — компоненты вектора F^j , $j = 1, 2, \dots, m-1$; l_1, l_2, \dots, l_n — компоненты вектора l ; $g_1, g_2, \dots, g_{n-n_0}$ — компоненты вектора g . Введем составные векторы Y и Φ размерности N каждый с компонентами Y_1, Y_2, \dots, Y_N и $\Phi_1, \Phi_2, \dots, \Phi_N$ соответственно:

$$Y = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^{m-1} \\ y^m \end{bmatrix}, \quad \Phi = \begin{bmatrix} l(y^1, Q) \\ F_1(y^1, y^2, Q) \\ \vdots \\ F^{m-1}(y^{m-1}, y^m, Q) \\ g(y^m, Q) \end{bmatrix}.$$

При этом система (10) записывается в виде (2). В дальнейшем под системой (2) будем подразумевать (10).

Предположим, что система $\Phi(Y, Q) = 0$ определяет гладкую пространственную кривую, задаваемую вектор-функцией $Y = Y(Q)$. Пусть по-прежнему Φ_Y — матрица якобиана, Φ_Q — вектор производных по параметру Q . В силу предположения ранг матрицы $[\Phi_Y, \Phi_Q]$ размеров

$N \times (N + 1)$ всегда равен N . Если $Y_0 = Y(Q_0)$ — решение (2) при $Q = Q_0$ и $\det \Phi(Y_0, Q_0) \neq 0$, то решение (2) может быть продолжено по параметру Q в некоторой окрестности Q_0 .

Обозначим через X вектор размерности N с компонентами $Y_1, Y_2, \dots, \dots, Y_{k-1}, Y_{k+1}, \dots, Y_N, Q$, и пусть $\lambda = Y_k = y_{j_*}^{i_*}, k = n(i_* - 1) + j_*, 1 \leq i_* \leq m, 1 \leq j_* \leq n$. Тогда краевой задаче (4) соответствует система трансцендентных уравнений вида (8) с параметром λ , причем эта система определяет гладкую пространственную кривую, задаваемую вектор-функцией $X = X(\lambda)$. Если $X_0 = X(\lambda_0)$ — решение системы $\Phi(X, \lambda) = 0$ при $\lambda = \lambda_0$ и $\det \Phi(X_0, \lambda_0) \neq 0$, то решение системы может быть продолжено по параметру λ в некоторой окрестности точки λ_0 . Если, кроме того, $Q = Q_0$ соответствует $\lambda = \lambda_0$, то $X = X(\lambda)$ является продолжением решения (2) при $Q = Q_0$.

Таким образом, идея параметризации применительно к системе (2) состоит в том, что любая компонента вектора Y и параметр Q в равной мере могут служить текущим параметром системы, т. е. использоваться для продолжения решения. В случае, когда (2) описывает гладкую пространственную кривую, причем вектор-функция $Y(Q)$ не обязана быть однозначной, эта кривая может быть построена численно методом продолжения решения, сочетающим метод Ньютона — Канторовича и параметризацию. Действительно, вдоль пространственной кривой ранг матрицы $[\Phi_Y, \Phi_Q]$ всегда равен N и, следовательно, мы всегда можем выбрать среди компонент вектора Y и Q параметр для продолжения решения (2), исходя из любой точки гладкой пространственной кривой.

Обозначим через μ текущий параметр системы, т. е.

$$\mu = \begin{cases} Q, & \text{если } \det \Phi_Y \neq 0, \\ \lambda, & \text{если } \det \Phi_X \neq 0. \end{cases}$$

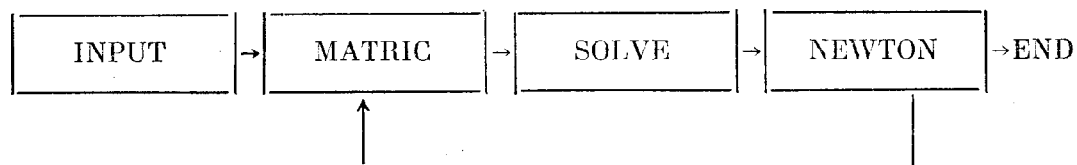
Подчеркивая регулярное применение параметризации, запишем (2) в виде

$$\Phi(S, \mu) = 0, \quad (11)$$

где

$$S = \begin{cases} Y, & \text{если } \mu = Q, \\ X, & \text{если } \mu = \lambda, \end{cases}$$

так что $\det \Phi_S \neq 0$. При описании решения (11) мы будем придерживаться следующей блок-схемы:



Рассмотрим содержание блоков с указанными названиями, если μ — заданный параметр, и при этом значении требуется найти решение (11).

Пусть $S = S^0$ — начальное приближение решения (11) при $\mu = \mu_0$. В силу определения вектора S это означает задание совокупности элементов $Y_1^0, Y_2^0, \dots, Y_N^0, Q$, в которой первые N элементов являются компонентами вектора $Y = Y_0$. Обозначим через k_* номер элемента этой последовательности, выбранного в качестве параметра μ системы (11), $k_* = 1, 2, \dots, N + 1$. По известным Y^0 и Q_0 (или, что то же самое, по известным S^0 и μ_0) вычислим элементы матрицы $[\Phi_Y, \Phi_Q]$ и вектора Φ . Вектор-столбец матрицы $[\Phi_Y, \Phi_Q]$ с номером k_* представляет вектор Φ_{μ} . Далее, требуется сформировать матрицу Φ_S , вычеркнув столбец матрицы $[\Phi_Y, \Phi_Q]$ с номером k_* и обратиться к методу Ньютона — Канторовича для уточнения приближения S^0 . После того как решение (11)

найденно, мы имеем возможность определить вектор производных S_μ из системы

$$\Phi_S(S, \mu) S_\mu = -\Phi_\mu(S, \mu).$$

Приведем блок-схему 1 решения (11) при заданном μ по методу Ньютона — Канторовича (рис. 4)*. В блок-схеме используются следующие обозначения:

Y^0, Q_0 — начальное приближение решения (11),

k_* — индекс текущего параметра, определяющего вектор S^0 и μ_0 ,

ϵ_0 — положительный достаточно малый параметр, задающий условие выхода из итерационного процесса по методу Ньютона — Канторовича,

i_0 — счетчик итераций,

n_9 — ограничение на число итераций,

k_0 — параметр, определяющий переход к вычислению S_μ : $k_0 = 1$ при выполнении итераций; $k_0 = 2$ при завершении итераций.

На выходе блок-схемы — векторы S и S_μ .

Отметим V-образную структуру матрицы $[\Phi_Y, \Phi_Q]$ системы (2), причем Φ_Y — блочно-двухдиагональная матрица:

$$[\Phi_Y, \Phi_Q] = \begin{bmatrix} l_{y^1} & & & & & l_Q \\ F_{y^1}^1 & F_{y^2}^1 & & & & F_Q^1 \\ & F_{y^2}^2 & F_{y^3}^2 & & & F_Q^2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ & & F_{y^{m-2}}^{m-2} & F_{y^{m-1}}^{m-2} & & F_Q^{m-2} \\ & & & F_{y^{m-1}}^{m-1} & F_{y^m}^{m-1} & F_Q^{m-1} \\ & & & & g_{y^m} & g_Q \end{bmatrix}$$

Здесь $l_{y^1}, g_{y^m}, F_{y^j}^{(i)}$ — блоки размеров $n_0 \times n$, $(n - n_0) \times n$, $n \times n$ соответственно. С учетом векторов Φ и Φ_Q размерности N каждый, $N = mn$, общее число элементов, вовлекаемых в вычислительный процесс, равно $2(n^2 + 1)N - n^2$. Естественно, что для снижения трудоемкости численной реализации метода Ньютона — Канторовича желателен учет разреженности матрицы Φ_S , образующейся вычеркиванием столбца матрицы $[\Phi_Y, \Phi_Q]$ с номером k_* , $k_* = 1, 2, \dots, N + 1$.

Приведем формулы, по которым вычисляются элементы блочных матриц $F_{y^i}^i, F_{y^{i+1}}^i$ и компоненты вектора F_Q^i . Для простоты введем обо-

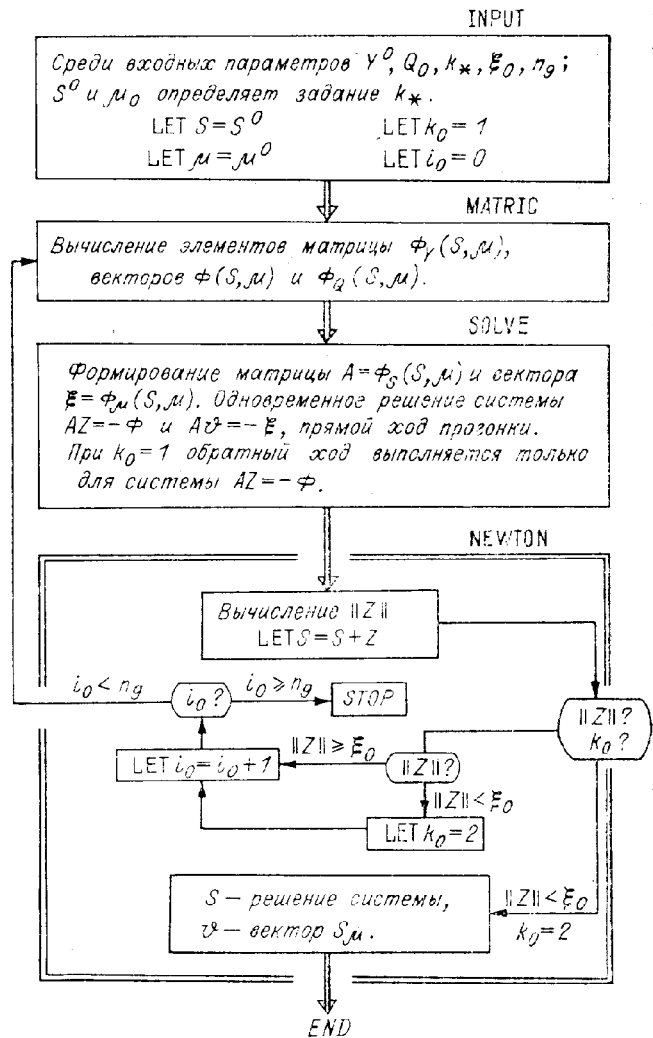


Рис. 4. Блок-схема 1.

* В дальнейшем будут использоваться элементы BASIC — одного из наиболее простых и распространенных языков программирования. Это вполне отвечает простоте структуры описываемых программ. При изображении блок-схем мы стремились достичь наибольшей наглядности.

значения $u = y^i$, $v = y^{i+1}$, $w = \bar{y}^i$, $h = h_i$ и опустим полное перечисление аргументов функции f . Имеем

$$F^i = u - v + \frac{h}{6} [f(u) + 4f(w) + f(v)],$$

где $w = \frac{1}{2}(u + v) + \frac{h}{8} [f(u) - f(v)]$. Отсюда

$$F_u^i = I + \frac{h}{6} [f_y(u) + 4f_y(w) w_u],$$

$$F_v^i = -I + \frac{h}{6} [f_y(v) + 4f_y(w) w_v],$$

где f_y , w_u , w_v — матрицы якобиана,

$$w_u = \frac{1}{2}I + \frac{h}{8}f_y(u), \quad w_v = \frac{1}{2}I - \frac{h}{8}f_y(v).$$

Целесообразно исключить w_u и w_v из выражений для F_u^i и F_v^i , после чего эти выражения принимают вид

$$F_u^i = I + \frac{h}{6} \left[f_y(u) + 2f_y(w) + \frac{h}{2} f_y(w) f_y(u) \right],$$

$$F_v^i = -I + \frac{h}{6} \left[f_y(v) + 2f_y(w) - \frac{h}{2} f_y(w) f_y(v) \right].$$

В принятых обозначениях

$$F_Q^i = \frac{h}{6} [f_Q(u) + 4f_Q(w) + 4f_y(w) w_Q + f_Q(v)], \quad w_Q = \frac{h}{8} [f_Q(u) - f_Q(v)],$$

где f_Q и w_Q — векторы производных f и w по параметру Q .

Кроме того, матрица $[\Phi_Y, \Phi_Q]$ содержит «нестандартные» блоки l_y^i g_{ym} и векторы производных l_Q^i , g_Q . Их выражения, наряду с выражениями l , g , f , f_y и f_Q определяются конкретной формулировкой краевой задачи (1).

§ 3. ПРОДОЛЖЕНИЕ РЕШЕНИЯ ПО ПАРАМЕТРУ И ПАРАМЕТРИЗАЦИЯ

Пусть k_* — номер элемента последовательности Y_1, Y_2, \dots, Y_N, Q , являющегося текущим параметром μ системы (11), и пусть при $\mu = \mu_0$ найдены решение системы $S(\mu_0)$ и вектор производных $S_\mu(\mu_0)$. В силу гладкости пространственной кривой, задаваемой системой (2), решение (11) может быть продолжено по параметру μ в некоторой окрестности точки μ_0 .

Рассматриваемый вариант продолжения решения состоит в следующем. Выберем достаточно малый шаг по μ , равный H . Тогда с погрешностью порядка H^2 в качестве начального приближения решение системы

$$\Phi(S, \mu_0 + H) = 0 \tag{12}$$

может быть взят вектор

$$S^0 = S(\mu_0) + HS_\mu(\mu_0). \tag{13}$$

При решении (12) по методу Ньютона — Канторовича, представленного блок-схемой 1, вводится ограничение на число итераций, за которое норма вектора поправки Z должна быть меньше ϵ_0 . Если число итераций i_0 достигло заданного значения n_8 , а $\|Z\| \geq \epsilon_0$, то начальное приближение S^0 можно улучшить за счет уменьшения шага H по параметру, например, в два раза. Тогда итерационный процесс решения системы

$$\Phi(S, \mu_0 + H/2) = 0 \tag{14}$$

начинается в приближения

$$S^0 = S(\mu_0) + \frac{H}{2} S_\mu(\mu_0). \quad (15)$$

Новый шаг делится пополам до тех пор, пока условие $\|Z\| \leq \epsilon_0$ не будет выполняться за число итераций $i_0 < n_8$. В этом отношении блок-схема 1 с параметром n_9 ограничения на число итераций описывает итерационный процесс в стартовой позиции построения пространственной кривой.

Естественно, что этим не исчерпывается проблема выбора шага по параметру. Понятие «достаточно малый шаг» остается большей частью предметом численного эксперимента, в котором разумным образом должны задаваться как ϵ_0 , n_8 , норма вектора поправки, так и формулироваться условие выхода из итерационного процесса (например, при $\|Z\|_c < \epsilon_0$, или $\|Z\|_c / \|S\|_c < \epsilon_0$). В целях программной адаптации текущего шага по параметру при численном построении пространственной кривой вводятся такие параметры, как минимально допустимый шаг по μ , максимально допустимый шаг, «стандартный» шаг, а также параметр «разгона» шага по μ , за счет которого регулярно предпринимается попытка постепенно увеличить шаг до максимального значения.

Пусть μ_0 — стартовое значение параметра μ , с которого начинается численное построение пространственной кривой; $S^{[1]} = S(\mu_1)$, $S_\mu^{[1]} = S_\mu(\mu_1)$, где $\mu_1 = \mu_0$. И пусть $S^{[2]} = S(\mu_2)$ — решение (11) при $\mu = \mu_2 = \mu_1 + H$, $S_\mu^{[2]} = S_\mu(\mu_2)$. Для дальнейшего продолжения решения системы по параметру μ выберем новый шаг H , $\text{sign}(H) = \text{sign}(\mu_2 - \mu_1)$, и будем искать решение системы

$$\Phi(S, \mu_2 + H) = 0, \quad (16)$$

учитывая при построении начального приближения решения S^0 векторы $S^{[1]}$, $S_\mu^{[1]}$, $S^{[2]}$, $S_\mu^{[2]}$. Воспользуемся вновь кубической параболой Эрмита:

$$S^0 = (1 - t)S^{[1]} + tS^{[2]} + t(1 - t)[(1 - t)R^{[1]} + tR^{[2]}]. \quad (17)$$

Здесь $t = 1 + H/H_0$, $H_0 = \mu_2 - \mu_1$, $H = \mu - \mu_2$, $t > 1$, $R^{[1]} = S^{[1]} - S^{[2]} + H_0 S_\mu^{[1]}$, $R^{[2]} = S^{[2]} - S^{[1]} - H_0 S_\mu^{[2]}$. В таком виде S^0 приближает решение (16) с погрешностью порядка H^4 . Значение H будем задавать согласно правилу

$$H = \begin{cases} \epsilon_2 H_0, & \text{если } \epsilon_2 |H_0| < \epsilon_3 H_*, \\ \epsilon_3 H_* \text{sign}(H_0), & \text{если } \epsilon_2 |H_0| \geq \epsilon_3 H_*, \end{cases} \quad (18)$$

где H_* — модуль эталонного шага по μ ; ϵ_2 — параметр разгона, $\epsilon_2 > 1$, $\epsilon_3 H_*$ — ограничение на величину шага по μ , $\epsilon_3 \geq 1$. В стартовой позиции шаг по μ выбирается пропорциональным $H_* \text{sign}(\omega)$, так что знак ω определяет изменение μ в сторону уменьшения или увеличения μ_0 . При этом начальное приближение решения (12) задается в виде (13).

После того как вектор S , являющийся решением (16), и вектор S_μ найдены, значения μ_2 и векторов $S^{[2]}$, $S_\mu^{[2]}$ присваиваются μ_1 , $S^{[1]}$ и $S_\mu^{[1]}$ соответственно, а μ_2 , $S^{[2]}$ и $S_\mu^{[2]}$ присваиваются значения μ , S и S_μ . Текущий шаг H по μ выбирается по правилу (18). Затем вновь рассматривается система (16) с начальным приближением решения в виде (17), и т. д.

Как и в случае (12), (13), полезно ввести ограничение на число итераций, за которое $\|Z\|$ становится меньше ϵ_0 , предусмотрев процедуру типа (14), (15), уточняющую начальное приближение S^0 (см. (17)) путем деления шага H пополам. Кроме того, вводится ограничение на величину шага H снизу: $|H| > \epsilon_1 H_*$, где ϵ_1 — достаточно малый параметр. При нарушении этого условия вычислительный процесс прекращается.

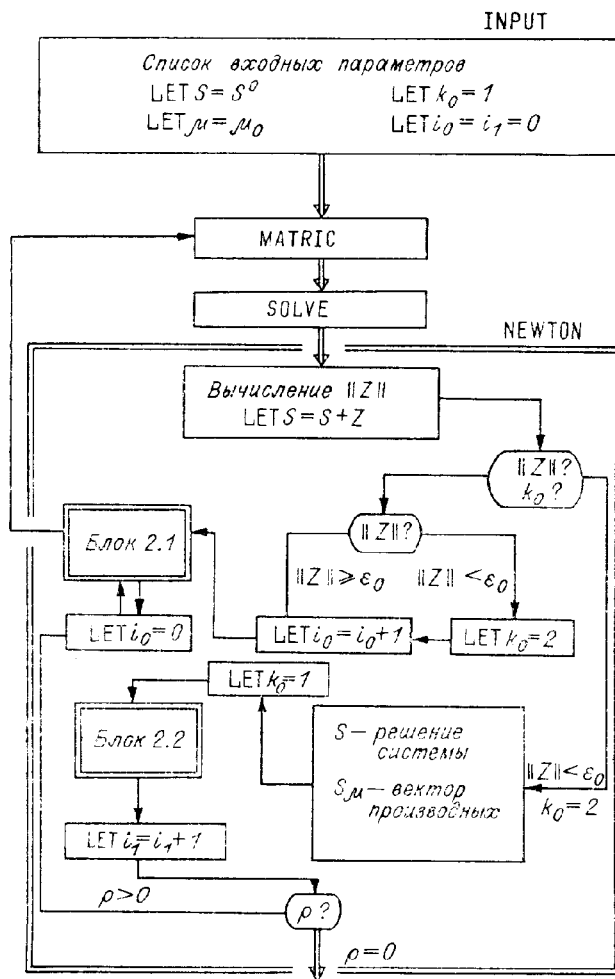


Рис. 5. Блок-схема 2.

таким же, как и в блок-схеме 1. Блок INPUT пополняется следующим списком параметров:

- H_* — модуль эталонного шага по μ ,
- ω — параметр, определяющий направление изменения стартового значения μ : $\omega = +1$ — в сторону увеличения, $\omega = -1$ — в сторону уменьшения,
- $\varepsilon_1, \varepsilon_3$ — параметры ограничения текущего шага H по μ : $\varepsilon_1 H_* < |H| \leq \varepsilon_3 H_*$,
- ε_2 — параметр «разгона» текущих шагов: последующий шаг по модулю берется равным предыдущему, умноженному на ε_2 , $\varepsilon_2 \geq 1$,
- n_9 — ограничение на число итераций при стартовом значении μ ,
- n_8 — ограничение на число итераций при текущих значениях μ .

Кроме того, вводятся счетчик шагов i_1 по параметру μ и признак p окончания работы программы: при $p = 0$ программа прекращает работу, в противном случае ($p > 0$) выполняется очередной шаг по параметру μ .

На рис. 6 и 7 приведены блок-схемы блоков 2.1 и 2.2, входящих в блок NEWTON. В блоке 2.1 формируется начальное приближение S^0 согласно (13), (17) и адаптация текущего шага H как следствие ограничений на число итераций. В блоке 2.2 запоминаются параметры, определяющие S^0 , и задается текущий шаг H .

Рассмотрим теперь продолжение решения по параметру с использованием параметризации. Пусть по-прежнему μ — текущий параметр системы (11) с индексом k_* , $k_* = 1, 2, \dots, N+1$, и пусть при $\mu = \mu_0$ найдены векторы S и S_μ . Тем самым при $\mu = \mu_0$ известны компоненты

Y_1, Y_2, \dots, Y_N вектора Y и параметр Q , а также компоненты $\frac{dY_1}{d\mu}, \frac{dY_2}{d\mu}, \dots, \frac{dY_N}{d\mu}$ вектора производных Y_μ и $\frac{dQ}{d\mu}$. Очевидно, что наряду с μ мы можем использовать i -ю компоненту вектора Y либо Q в качестве теку-

Заметим, что подход к проблеме продолжения решения как к задаче Коши типа (3)

$$S_\mu = -[\Phi_S(S, \mu)]^{-1} \Phi_\mu(S, \mu),$$

$$S|_{\mu=\mu_0} = S(\mu_0),$$

привлекателен прежде всего тем, что позволяет обратиться для ее решения к одной из стандартных программ. Однако следует иметь в виду, что в данном случае от решения задачи Коши фактически требуется не аппроксимация решения, а предсказание приемлемого начального приближения решения (11). Поэтому и здесь (при всех очевидных достоинствах такого подхода) возможны ситуации, в которых эффективность метода снижается. Например, задача Коши может оказаться жесткой с вытекающим отсюда ограничением на шаг интегрирования, связанным с аппроксимацией решения.

На рис. 5 приведена блок-схема 2 продолжения решения (11) по параметру μ . Содержание блоков MATRIC и SOLVE осталось

щего параметра системы, если $\frac{dY_j}{d\mu} \neq 0$, либо $\frac{dQ}{d\mu} \neq 0$. Напомним, что пространственная кривая, определяемая системой (2), по предположению не имеет особых точек.

Выберем в качестве нового текущего параметра α системы наиболее быстро меняющийся элемент совокупности Y_1, Y_2, \dots, Y_N, Q . Пусть

$$c = \max \left(\left| \frac{dY_1}{d\mu} \right|, \left| \frac{dY_2}{d\mu} \right|, \dots, \left| \frac{dY_N}{d\mu} \right|, \left| \frac{dQ}{d\mu} \right| \right).$$

Тогда α и производная α_μ определяется следующим образом:

$$\begin{aligned} \alpha = Y_2, \quad \alpha_\mu = \frac{dY_i}{d\mu}, \quad \left| \frac{dY_i}{d\mu} \right| = c, \\ \alpha = Q, \quad \alpha_\mu = \frac{dQ}{d\mu}, \quad \left| \frac{dQ}{d\mu} \right| = c. \end{aligned} \quad (19)$$

При этом $\alpha = \mu$, $\alpha_\mu = 1$, если $c = 1$. Чтобы воспользоваться описанной ранее процедурой продолжения решения по текущему параметру, найдем производные решения по α . Имеем

$$\begin{aligned} \frac{dY_1}{d\alpha} = \frac{1}{\alpha_\mu} \frac{dY_1}{d\mu}, \quad \frac{dY_2}{d\alpha} = \frac{1}{\alpha_\mu} \frac{dY_2}{d\mu}, \quad \dots, \\ \frac{dY_N}{d\alpha} = -\frac{1}{\alpha_\mu} \frac{dY_N}{d\mu}, \quad \frac{dQ}{d\alpha} = \frac{1}{\alpha_\mu} \frac{dQ}{d\mu}. \end{aligned} \quad (20)$$

В силу определения α

$$\max \left(\left| \frac{dY_1}{d\alpha} \right|, \left| \frac{dY_2}{d\alpha} \right|, \dots, \left| \frac{dY_N}{d\alpha} \right|, \left| \frac{dQ}{d\alpha} \right| \right) = 1.$$

Следовательно, применение параметризации (19) ограничивает рост производных решения системы по текущему параметру.

В дальнейшем условимся обозначать через \tilde{S} вектор, образованный из N элементов совокупности Y_1, Y_2, \dots, Y_N, Q в результате параметризации (19), (20) решения системы (11), \tilde{S}_α — вектор производных по новому параметру системы α . Таким образом, тождество $\Phi(S, \mu) \equiv 0$ после параметризации переписывается в виде $\Phi(\tilde{S}, \alpha) \equiv 0$. При этом, если индекс k_* элемента μ не совпадает с индексом \tilde{k}_* элемента α , то μ является компонентой вектора \tilde{S} , а производная $\mu_\alpha = 1/\alpha_\mu$ — компонентой вектора \tilde{S}_α .

Для описания продолжения решения по параметру с применением параметризации мы можем вновь воспользоваться блок-схемой 2 (рис. 8), в которой изменено лишь содержание блока 2.2. Пополним список обозначений в связи с параметризацией:

- \tilde{k}_* — индекс параметра α ,
- \tilde{H}_* — эталонный шаг по α ,
- α_2 — значение α , соответствующее $\mu = \mu_2$, $\tilde{S}^{[2]}$, $\tilde{S}_\alpha^{[2]}$ — векторы \tilde{S} и \tilde{S}_α при $\alpha = \alpha_2$.

Для удобства описания выделены блоки 2.2.1 и 2.2.2 (рис. 9) и рис. 10 соответственно). В блок-схему блока 2.2.1 включена процедура согласования знака шага по параметру α со знаком шага по параметру

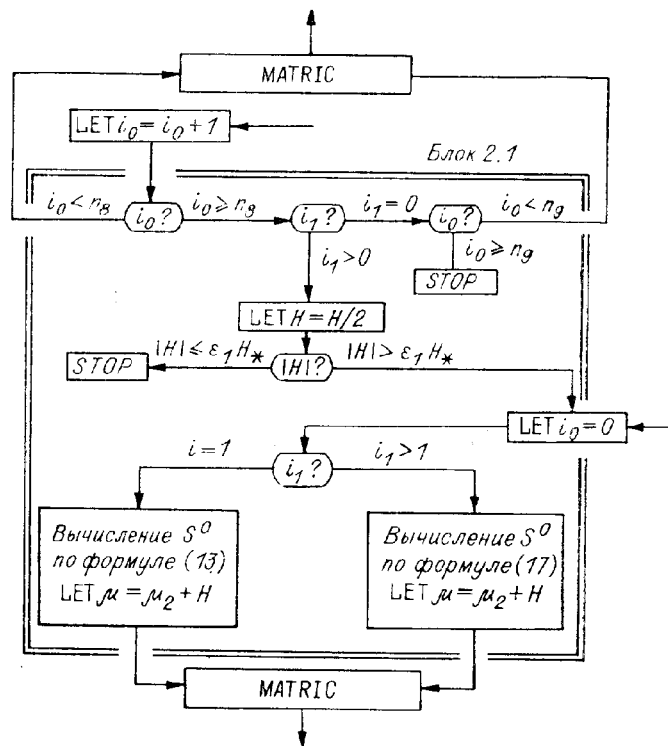


Рис. 6. Блок-схема блока 2.1.

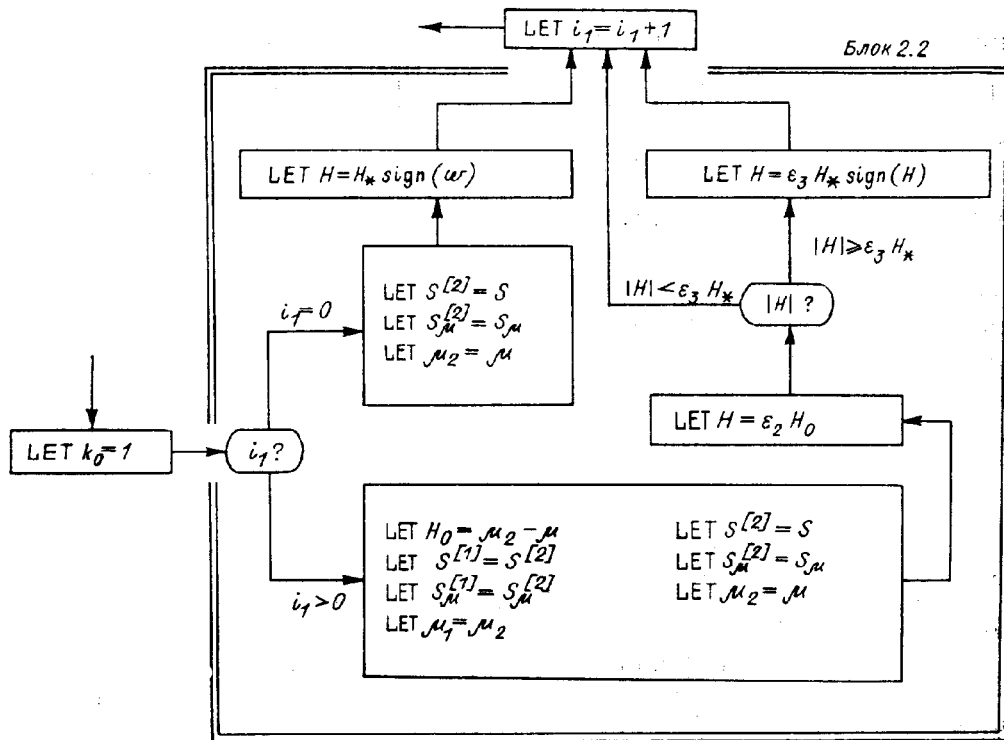


Рис. 7. Блок-схема 2.2.

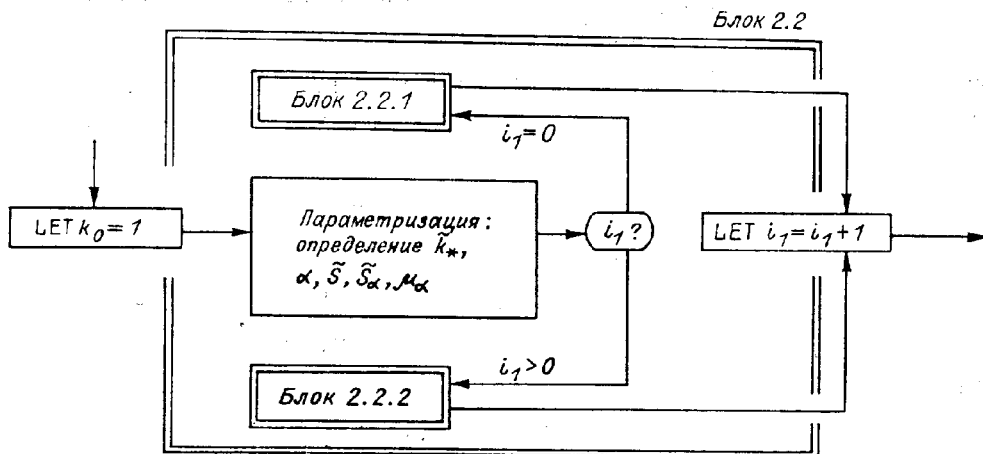


Рис. 8. Блок-схема блока 2.2.

и в стартовой позиции. Здесь для конкретности модуль шага по α взят равным \tilde{H}_* .

В том случае, когда рассматривается система трансцендентных уравнений общего вида типа (2), т. е. как правило, с заполненной матрицей якобиана Φ_Y , текущий параметр системы может быть найден иным

способом. Пусть по-прежнему ранг матрицы $[\Phi_Y, \Phi_Q]$ равен N . Воспользуемся методом Гаусса с выбором главного элемента по матрице для выявления базисных столбцов. Тогда столбец с номером \tilde{k}_* матрицы $[\Phi_Y, \Phi_Q]$ по завершению прямого хода метода Гаусса окажется в позиции $(N + 1)$ -го столбца. При обратном ходе определяются производные решения системы по параметру α . Однако применение этого метода параметризации к системе (2) с разреженной матрицей Φ_Y будет неэффективным, особенно при больших N , поскольку при этом игнорируется структура матрицы Φ_Y .

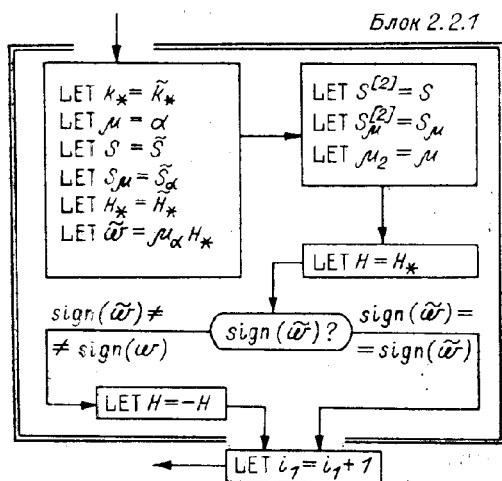


Рис. 9. Блок-схема блока 2.2.1.

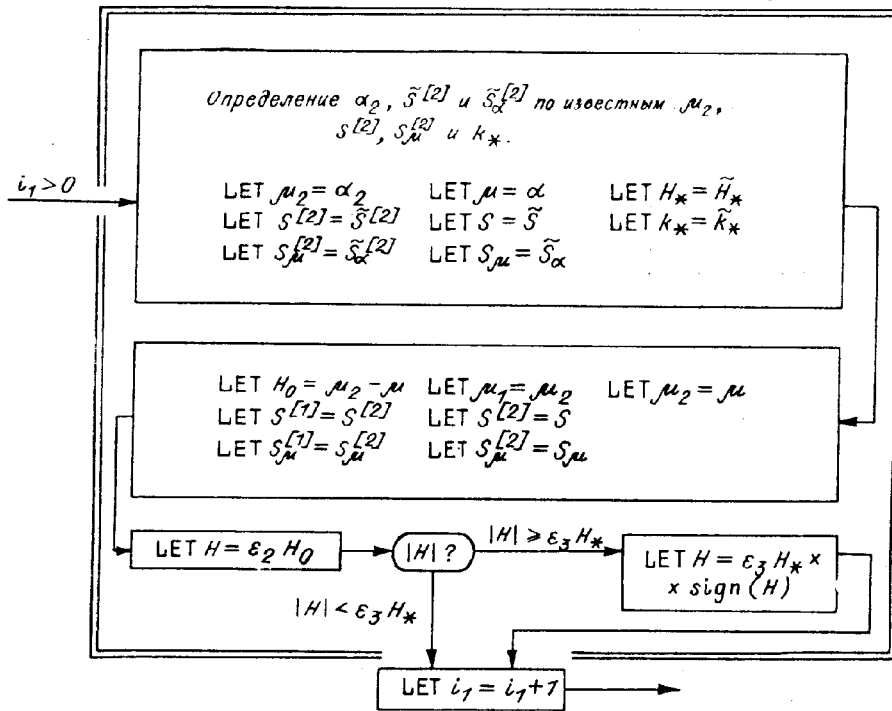


Рис. 10. Блок-схема блока 2.2.2.

В предлагаемом способе текущий параметр системы определяется по формулам (19). Поэтому матрица $\Phi_s(S, \mu)$, образующаяся вычеркиванием столбца с номером k_* матрицы $[\Phi_Y, \Phi_Q]$, остается разреженной. В следующем параграфе мы рассмотрим экономичный метод решения системы линейных алгебраических уравнений

$$\Phi_s(S^0, \mu)Z = -\Phi(S^0, \mu) \tag{21}$$

относительно вектора Z , уточняющего приближение S^0 решения системы (11).

§ 4. ОПИСАНИЕ БЛОКА SOLVE

Пусть по заданному приближению решения системы (11), текущий параметр μ которой имеет индекс k_* , найдены элементы блоков матрицы Φ_Y , векторов Φ_Q и Φ . Для решения системы (21) воспользуемся методом Гаусса с выбором главного элемента по столбцу, учитывающим структуру матрицы Φ_s . При этом прямой ход метода Гаусса — приведение Φ_s к верхней треугольной матрице $\tilde{\Phi}_s$ — сопровождается одновременным преобразованием как вектора Φ , так и вектора Φ_μ . Обратный ход определяет решение системы (21), а также решение системы

$$\Phi_s(S^0, \mu)S_\mu = -\Phi_\mu(S^0, \mu), \tag{22}$$

если итерационный процесс для системы (11) сошелся.

Возвращаясь к определению Y как составного вектора, образованного из векторов $y^i, i = 1, 2, \dots, m$, с компонентами $y_1^i, y_2^i, \dots, y_n^i$, будем характеризовать текущий параметр системы (11) двумя индексами i_* и j_* : если $i_* = m + 1, j_* = 0$, то $\mu = Q$, что соответствует $k_* = N + 1 = = mn + 1$; если $1 \leq i_* \leq m, 1 \leq j_* \leq n$, то $\mu = y_{j_*}^{i_*}$, т. е. $k_* = n(i_* - 1) + j_*$.

Чтобы охватить все варианты структуры матрицы Φ_s , возникающие при вычеркивании столбца матрицы $[\Phi_Y, \Phi_Q]$ с номером k_* , достаточно рассмотреть четыре случая задания i_* : а) $i_* = m + 1$, б) $i_* = 1$, в) $i_* = m$ и д) $1 < i_* < m$.

При реализации прямого хода прогонки методом Гаусса или, как еще говорят, прямого хода прогонки, выполняемого за m шагов, мы будем использовать вспомогательный прямоугольный массив w размеров

Представим вектор Z в виде составного вектора, образованного из векторов Z^j , $j = 1, 2, \dots, m$, с компонентами $Z_1^j, Z_2^j, \dots, Z_n^j$. Согласно схеме (25) на первом шагу обратного хода прогонки ($i = m$) определяется вектор Z^m из решения системы $A^{[m]}Z^m = -F^{[m]}$. Дальнейшие шаги, $i = m - 1, m - 2, \dots, 2, 1$, последовательно определяют векторы $Z^{m-1}, Z^{m-2}, \dots, Z^2, Z^1$ по рекуррентным формулам $A^{[i]}Z^i = -F^{[i]} - B^{[i]}Z^{i+1}$.

(b) Пусть $i_* = 1, 1 \leq j_* \leq n$. При этом $k_* = j_*$, т. е. $\mu = y_{j_*}^1$, а в матрице Ω (см. (23)) вычеркивается столбец с номером j_* . В результате матрица $\Omega^{[0]} = [\Phi_s, \Phi, \Phi_\mu]$ в схеме прямого хода прогонки (24) имеет вид

$$\Omega^{[0]} = \left[\begin{array}{c|cc} & \Phi_s & \Phi & \Phi_\mu \\ \hline GG & & G & G \\ GG & & G & G \\ AA & AAA & Q & F & A \\ AA & AAA & Q & F & A \\ AA & AAA & Q & F & A \\ & AAA & AAA & Q & F & O \\ & AAA & AAA & Q & F & O \\ & AAA & AAA & Q & F & O \\ & & AAA & AAA & Q & F & O \\ & & AAA & AAA & Q & F & O \\ & & AAA & AAA & Q & F & O \\ & & & GGG & G & G & O \end{array} \right].$$

Обратим внимание на задание массива W в стартовой позиции:

$$W = \left[\begin{array}{ccccccc} GG & 000 & 0 & G & G & G \\ GG & 000 & 0 & G & G & G \\ AA & AAA & 0 & Q & F & A \\ AA & AAA & 0 & Q & F & A \\ AA & AAA & 0 & Q & F & A \end{array} \right], \quad L_1 = 5, \quad L_2 = 9, \quad L_3 = 2.$$

Здесь в массив W искусственно включен нулевой столбец с целью единообразного описания прямого хода прогонки. Приведем выражения остальных матриц (24):

$$\Omega^{[1]} = \left[\begin{array}{c|cc} AA & AAA & Q & F & R \\ A & AAA & Q & F & R \\ \hline & XXX & 000 & X & X & X \\ & XXX & 000 & X & X & X \\ & XXX & 000 & X & X & X \\ W & AAA & AAA & Q & F & O \\ & AAA & AAA & Q & F & O \\ & AAA & AAA & Q & F & O \\ & & AAA & AAA & Q & F & O \\ & & AAA & AAA & Q & F & O \\ & & AAA & AAA & Q & F & O \\ & & & GGG & G & G & O \end{array} \right],$$

$$\Omega^{[2]} = \left[\begin{array}{c|cc} AA & AAA & Q & F & R \\ A & AAA & Q & F & R \\ & AAA & AAA & Q & F & R \\ & AA & AAA & Q & F & R \\ & A & AAA & Q & F & R \\ \hline & XXX & 000 & X & X & X \\ & XXX & 000 & X & X & X \\ & XXX & 000 & X & X & X \\ W & AAA & AAA & Q & F & O \\ & AAA & AAA & Q & F & O \\ & AAA & AAA & Q & F & O \\ & & & GGG & G & G & O \end{array} \right].$$

При $1 < i_* < m$ параметры L_1, L_2 и L_3 принимают значения:

$$\begin{aligned} L_1 &= n_0 + n, & L_2 &= 2n + 3, & L_3 &= n, & 1 \leq i < i_*, \\ L_1 &= n_0 + n, & L_2 &= 2n + 3, & L_3 &= n - 1, & i = i_*, \\ L_1 &= n_0 + n + 1, & L_2 &= 2n + 3, & L_3 &= n, & i_* < i < m, \\ & & L_1 &= n + 1, & L_2 &= n + 3, & L_3 &= n, & i = m. \end{aligned}$$

Матрица $\Omega^{[m]}$ вновь имеет вид (26), причем

$$\begin{aligned} L_4 &= n, & L_5 &= n, & 1 \leq i < i_* - 2, \\ L_4 &= n, & L_5 &= n - 1, & i = i_* - 1, \\ L_4 &= n - 1, & L_5 &= n, & i = i_*, \\ L_4 &= n, & L_5 &= n, & i_* < i \leq m. \end{aligned}$$

Поэтому описание обратного хода прогонки, состоящего в решении системы (27), совпадает со случаем (b), включая определение Z_j^{i*} (см. (28)).

§ 5. АДАПТАЦИЯ СЕТКИ

При продолжении по параметру решения системы трансцендентных уравнений, аппроксимирующей дифференциальную краевую задачу, возникает, вообще говоря, проблема адаптации первоначальной заданной сетки по x к изменению параметра. Рассмотрим один простой способ вариации узлов сетки, позволяющий повысить точность численного решения дифференциальной краевой задачи во многих случаях. Подчеркивая эвристический характер рассуждений, мы будем называть его правилом адаптации сетки с заданным числом узлов. В нем заимствуется процедура выбора узлов сетки при сплайн-интерполяции функции с заданной точностью.

Пусть, например, функция $u(x)$, принадлежащая классу C^4 на отрезке $[a, b]$, приближается эрмитовым кубическим сплайном. Требуется подобрать узлы интерполяции таким образом, чтобы абсолютная величина погрешности приближения не превышала заданного значения ε . Таковыми являются узлы сетки

$$a = x_1 < x_2 < \dots < x_m = b, \quad h_i = x_{i+1} - x_i \quad (29)$$

при выполнении равенств

$$\varepsilon = \frac{h_i^4}{384} \left| \frac{d^4 u}{dx^4}(\theta_i) \right|, \quad \theta_i \in [x_i, x_{i+1}],$$

или

$$h_i \left| \frac{d^4 u}{dx^4}(\theta_i) \right|^{1/4} = (384\varepsilon)^{1/4}. \quad (30)$$

Пусть теперь $u(x)$ — компонента вектор-функции $y(x)$, представляющей численное решение краевой задачи (1) с использованием первоначально заданной сетки по x :

$$a = \xi_1 < \xi_2 < \dots < \xi_m = b, \quad \delta_i = \xi_{i+1} - \xi_i. \quad (31)$$

Предположим, что в соответствии с порядком аппроксимации дифференциальной краевой задачи системой трансцендентных уравнений абсолютная величина погрешности, связанная с $u(x)$, пропорциональна $\delta^4 \left| \frac{d^4 u}{dx^4} \right|$, $\delta = \max_i \delta_i$. Если же построенное численное решение оказалось достаточно близким к точному, то, используя приближенное выражение чет-

вертой производной, можно попытаться улучшить его точность путем пересчета задачи (1) на вновь сформированной сетке (29) с варьированными узлами (31). Исходя из равенств

$$\varepsilon = Ch_i^4 \left| \frac{d^4 u}{dx^4}(\theta_i) \right|, \quad \theta_i \in [x_i, x_{i+1}],$$

выполняющихся для каждого из отрезков $[x_i, x_{i+1}]$, где $x_i, i = 1, 2, \dots, m-1$, подлежат определению, а $C > 0$ — некоторая постоянная, составим сумму

$$C^{\frac{1}{4}} \sum_{i=1}^{m-1} h_i \left| \frac{d^4 u}{dx^4}(\theta_i) \right|^{\frac{1}{4}} = (m-1) \varepsilon^{\frac{1}{4}} \approx C^{\frac{1}{4}} \int_a^b \left| \frac{d^4 u}{dx^4}(x) \right|^{\frac{1}{4}} dx.$$

Отсюда следует равенство, аналогичное (30):

$$\int_{x_i}^{x_{i+1}} \left| \frac{d^4 u}{dx^4}(x) \right|^{\frac{1}{4}} dx = \frac{1}{m-1} \int_a^b \left| \frac{d^4 u}{dx^4}(x) \right|^{\frac{1}{4}} dx. \quad (32)$$

Подчинив узлы сетки (29) условиям (32), мы вправе ожидать, что численное решение краевой задачи на этой сетке будет уже характеризоваться распределением погрешности, близким к равномерному (хотя значение ε остается при этом неизвестным). Здесь мы можем сослаться лишь на численный эксперимент, говорящий о том, что использование сеток суказанным свойством равномерности распределения погрешности приводит (как правило) к существенному повышению точности численного решения. (В другом варианте $u(x) = \|y(x)\|$.)

На практике удобно заменить (32) близким по смыслу условием. Пусть

$$\varphi(z) = \int_a^z \left| \frac{d^4 u}{dx^4}(x) \right|^{\frac{1}{4}} dx, \quad a \leq z \leq b.$$

Напомним, что реально подынтегральная функция определяется из численного решения краевой задачи (1) на сетке (31). Тогда правило выбора i -го узла сетки (29) состоит в решении следующего уравнения относительно $x_i, i = 2, 3, \dots, m-1$:

$$\varphi(x_i) = \frac{i-1}{m-1} \varphi(b). \quad (33)$$

Очевидно, в силу монотонности $\varphi(z)$ уравнение (33) имеет единственное решение.

Считая сетку (31) достаточно густой, будем предполагать в дальнейшем, что распределение погрешности решения краевой задачи достаточно хорошо учитывается кусочно-постоянной функцией, определяемой на каждом из интервалов $[\xi_i, \xi_{i+1}]$ значением четвертой производной, вычисленной, например, в средней точке интервала. При этом графиком $\varphi(z)$ будет ломаная кривая, а задача построения сетки (29) согласно (33) сведется к отысканию точки пересечения ломаной с прямой, параллельной оси абсцисс. На рис. 11 дана схема формирования сетки (2) при $m = 5$. Здесь $\Delta = \varphi(b)/(m-1)$.

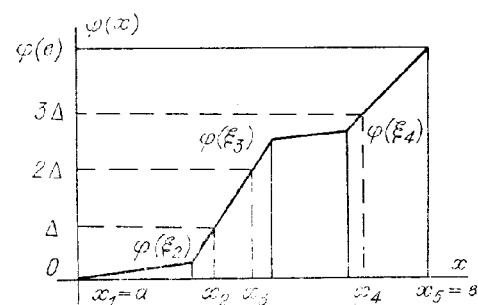


Рис. 11.

Пусть из решения краевой задачи в узлах сетки (31) известны функция $u(x)$, ее первая и вторая производные:

$$u = u_i, \quad \frac{du}{dx} = u'_i, \quad \frac{d^2 u}{dx^2} = u''_i \quad \text{при } \xi = \xi_i.$$

(Если $y(x)$ — решение (1), то $\frac{dy}{dx} = f$,

$\frac{d^2 y}{dx^2} = \frac{\partial f}{\partial x} + f_y f$, где f_y — матрица якобиана). Для приближенного выражения четвертой производной воспользуемся эрмитовым сплайном 5-й степени класса C^2 :

$$\begin{aligned} \xi_i < x < \xi_{i+1}, \quad x = \xi_i + \delta_i t, \quad \delta_i = \xi_{i+1} - \xi_i, \quad 0 \leq t \leq 1, \\ u(x) \approx (1-t)u_i + tu_{i+1} + t(1-t)[(1-t)a_i + tb_i] + \\ + t^2(1-t)^2[(1-t)\alpha_i + t\beta_i], \end{aligned}$$

где

$$\begin{aligned} a_i &= \delta_i u'_i + u_i - u_{i+1}, \quad b_i = u_{i+1} - u_i - \delta_i u'_{i+1}, \\ \alpha_i &= \frac{\delta_i^2}{2} u''_i + 2a_i - b_i, \quad \beta_i = \frac{\delta_i^2}{2} u''_{i+1} + 2b_i - a_i. \end{aligned}$$

Отсюда

$$\frac{d^4 u}{dx^4} \left(\frac{\xi_i + \xi_{i+1}}{2} \right) \approx c_i = \frac{12}{\delta_i^3} \left(u'_i - u'_{i+1} + \delta_i \frac{u''_i + u''_{i+1}}{2} \right).$$

Введем обозначение $\varphi_k = \varphi(\xi_k)$, так что

$$\begin{aligned} \varphi_{k+1} &= \varphi_k + \delta_k |c_k|^{1/4}, \quad k = 1, 2, \dots, m-1, \\ \varphi_1 &= 0, \quad \varphi_m = \varphi(b). \end{aligned}$$

Если $\varphi_k \leq i\Delta \leq \varphi_{k+1}$, $\Delta = \varphi_m / (m-1)$, $i = 1, 2, \dots, m-2$, то уравнение (33) принимает вид

$$\varphi_k + \frac{\varphi_{k+1} - \varphi_k}{\delta_k} (x_{i+1} - \xi_k) = i\Delta.$$

Таким образом, узел сетки x_{i+1} , соответствующий $\varphi(x_{i+1}) = i\Delta$, находится по формуле,

$$x_{i+1} = \xi_k + \delta_k \frac{i\Delta - \varphi_k}{\varphi_{k+1} - \varphi_k}, \quad \varphi_k \leq i\Delta < \varphi_{k+1}. \quad (34)$$

Остановимся на описании процедуры адаптации сетки при продолжении решения краевой задачи (1) по параметру, отмечая через i_1 количество шагов по текущим параметрам ($i_1 = 0$ в стартовой позиции). Пусть при $i_1 = n_7$, где n_7 — заданное целое число, найдено решение системы (11), определенной на сетке (31) по формулам (10) при известном значении текущего параметра μ , и требуется «подправить» узлы сетки ξ_k , $k = 1, 2, \dots, m$. Поскольку известны индексы i_* и j_* параметра μ , $1 \leq i_* \leq m+1$, $1 \leq j_* \leq n$, то компоненты вектора S и μ задают составной вектор Y , состоящий из векторов y^i , $i = 1, 2, \dots, m$, с компонентами y_j^i , $j = 1, 2, \dots, n$, и параметр Q . Обозначим через j_0 индекс компонент векторов y^i , $i = 1, 2, \dots, m$, значения которых мы будем принимать во внимание при вариации узлов сетки (31), подчинив вариации правилу (34) (применительно к обозначениям (32) $u(x) = y_{j_0}(x)$ — компонента вектор-функции $y(x)$ с номером j_0). В результате имеем сетку (29).

Затем краевая задача (1) пересчитывается на сетке (29) с использованием полученного ранее решения на сетке (31) в качестве начального приближения. Обратимся к системе (11), определенной на сетке (29). При этом, вообще говоря, меняется индекс i_* текущего параметра μ , если $1 < i_* < m+1$. Применительно к сетке (29) будет считать, что $i_* = k$, если $x_k \leq \xi_{i_*} \leq x_{k+1}$ и $\xi_{i_*} - x_k < x_{k+1} - \xi_{i_*}$. Если $x_{k+1} - \xi_{i_*} < \xi_{i_*} - x_k$, то $i_* = k+1$. Начальное приближение S^0 решения (11) задается следующим образом. На отрезке $[a_0 b]$ строится эрмитов кубический сплайн $\mathcal{P}(x)$ класс C^1 по известным в узлах сетки (31) значениям $y^i = y(\xi_i)$ и $(y^i)' = f(\xi_i, y^i, Q)$, так что $\mathcal{P}(\xi_i) = y^i$, $\frac{d\mathcal{P}}{dx}(\xi_i) = (y^i)'$. Значения вектор-

функции $\mathcal{P}(x)$ в узлах сетки (29) определяют новые векторы $y^i = \mathcal{P}(x_i)$, $i = 1, 2, \dots, m$, из которых составляем новый вектор Y , а следовательно, включив сюда Q , и S^0 . Заметим, что если $1 < i_* < m + 1$, то вместе с индексом i_* изменяется значение параметра μ , совпадающего со значением компоненты с номером j_* вектора $\mathcal{P}(x_{i_*})$.

Очередной после $i_1 = n_7$ шаг продолжения решения по параметру следует рассматривать как стартовый, положив $i_1 = 0$. При этом предполагается сохранить правило (18) выбора шага H по параметру μ , взяв в качестве H_0 приращение μ на предыдущем шагу, где применялась сетка (31). Так как знаки приращений компонент векторов y^1, y^m и Q не зависят от вариации узлов сетки, то это обстоятельство используется для контроля правильности выбора направления изменения найденного текущего параметра μ : ставится условие, согласно которому знак максимального по модулю приращения Δ_0 среди приращений упомянутых параметров $y_j^1, y_j^m, j = 1, 2, \dots, m$, и Q , связанных с сеткой (31), должен сохраниться при продолжении решения по параметру системы (11), определенной на сетке (29). Если максимальное приращение по модулю оказалось меньше заданного числа ε_4 , то попытка адаптации сетки принимается при $i_1 > n_7$.

Следует добавить, что при переходе к сетке (29) меняется набор модулей эталонных шагов H_{ij} , $1 \leq i \leq m, 1 \leq j \leq n$, заданных на сетке (31) для текущих параметров μ системы (11), в качестве которых могут быть выбраны компоненты y_j^i векторов $y^i, 1 \leq i \leq m, 1 \leq j \leq n$ и $Q, i = m + 1, j = 0$. (Ранее мы обозначили через H_* модуль эталонного шага текущего параметра μ , так что $H_{ij} = H_*$.) При этом, очевидно, значения $H_{ij}, H_{mj}, 1 \leq j \leq n$, и модуля эталонного шага для Q остаются неизменными. Остальные новые значения H_{ij} определяются на сетке (29) из простой линейной интерполяции.

Рассмотренное правило адаптации сетки дополняется еще одним полезным приемом. Если расстояние $x_{k+1} - x_k, k = 1, 2, \dots, m - 1$, между

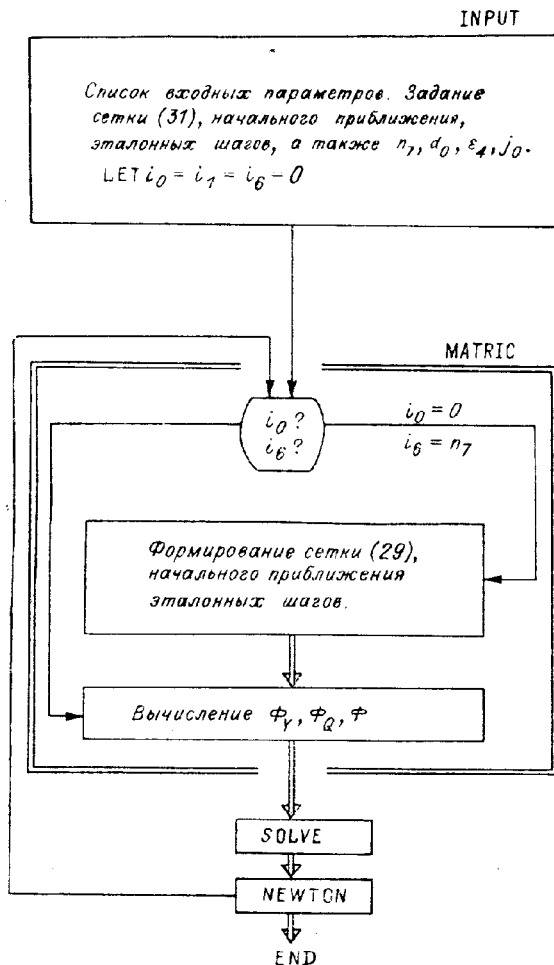


Рис. 12. Блок-схема 3.

Содержание блока NEWTON представлено на рис. 5, 6, 8, 13, 14.

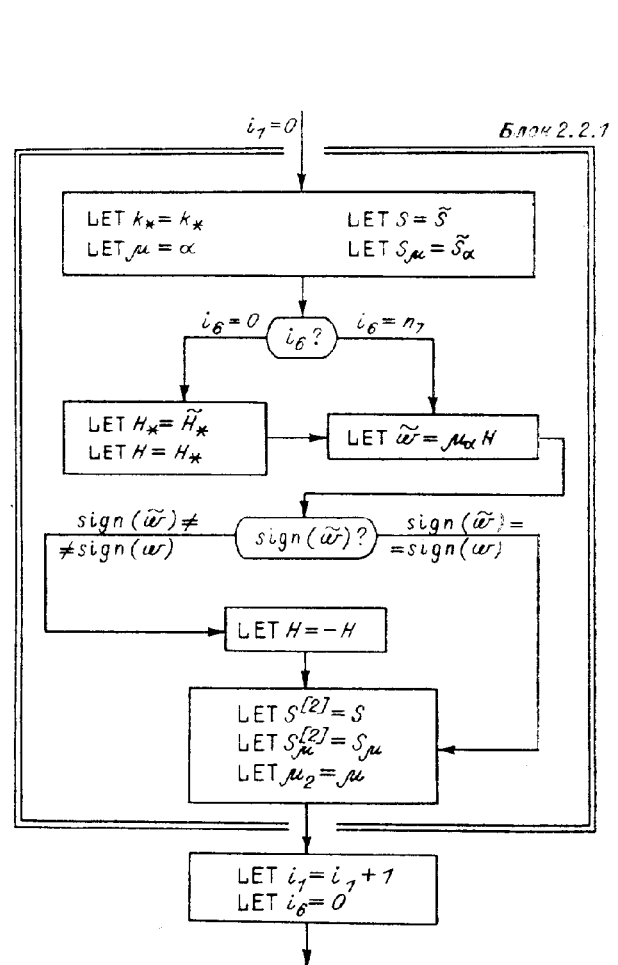


Рис. 13. Блок-схема блока 2.2.1.

узлами сетки (29) оказалось больше заданного d_0 , то вводятся дополнительные узлы путем равномерного разбиения отрезка $[x_k, x_{k+1}]$ на t частей, $t = 1 + \text{int}((x_{k+1} - x_k)/d_0)$, $d_0 < b - a$. В результате узлы сетки (29) удовлетворяют условию $x_{k+1} - x_k \leq d_0$. При этом меняется значение m , а вместе с ним и размерность системы (11). Дальнейшее описание процедуры адаптации сетки в этом случае повторило бы приведенное выше.

В связи с введением процедуры адаптации сетки список параметров блока INPUT пополняется параметрами $n_7, j_0, \Delta_0, d_0, \varepsilon_4$ и i_6 . Признак i_6 позволяет отличить действительно стартовую позицию продолжения решения по параметру ($i_1 = 0, i_6 = 0$, блок INPUT) от позиции, возникающей при обращении к процедуре адаптации сетки ($i_1 = 0, i_6 = n_7$, блок NEWTON). Сама процедура, оформленная в виде подпрограммы, содержится в блоке MATRIS. При этом к «нестандартным» блокам вычисления $l_{y1}, l_Q, l, g_{ym}, g_Q, g, f_v, f_Q$ и f добавляется блок вычисления f_x для определения $\frac{d^2y}{dx^2}$. Особенности процесса продолжения решения по параметру, сопровождающегося адаптацией сетки, схематично отражены на рис. 12—14. На рис. 13, 14 приведены блок-схемы модифицированных блоков 2.2.1 и 2.2.2 блока 2.2 NEWTON, описание которых дано на рис. 5.8—10.

§ 6. ПРОГРАММА BPR-Q

Имя BPR-Q присвоено программе решения нелинейной краевой задачи (1) способом, изложенным в § 2—5. Текст программы на языке BASIC дан в приложении к § 6. Благодаря простоте структуры программы требуются минимальные сведения о языке BASIC для прочтения текста. Поэтому использование BASIC в данном случае позволяет отвлечься от многих условностей программной реализации, имеющих место при обращении к другим языкам, и сосредоточить внимание на деталях обсуждаемого метода.

Блоки INPUT, MATRIS, SOLVE, NEWTON оформлены в виде программ, содержащих предложение CHAIN для сборки этих программ в программу BPR-Q. Отметим, что BPR-Q можно записать в виде одной программы простым объединением текстов последовательности TINPUT, MATRIS, SOLVE, NEWTON. При этом требуется заменить предложения CHAIN на предложения GOTO, передав управление на начало программы, имя которой указано в предложении CHAIN (не считая началом предложения COM), а также переписать номера предложений в возрастающей последовательности.

Формально имя BPR-Q присвоено INPUT, а собственно блок INPUT оформлен в виде подпрограммы BPR-Q. В подпрограмме содержится входная информация, задаваемая пользователем при решении конкретной задачи. На рис. 15 приведена блок-схема BPR-Q с перечислением подпрограмм. В тексте наименование подпрограмм содержится только в

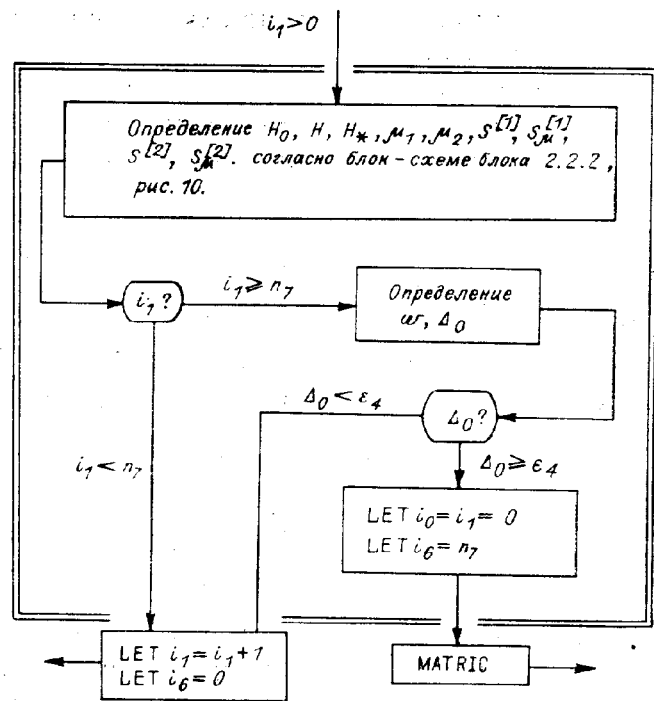


Рис. 14. Блок-схема блока 2.2.2.

предложениях REM и используется лишь для наглядности представления программы.

В блок-схему включена программа ERROR, в которой вычисляются невязки системы (9) при подстановке в нее найденного численного решения (1). Определим приближенное представление $y(x)$ задачи (1) в виде

$$x_i \leq x \leq x_{i+1}, h_i = x_{i+1} - x_i, x = x_i + h_i \tau, 0 \leq \tau \leq 1,$$

$$y(x) \approx \mathcal{P}_i(x) = (1 - \tau)y^i + \tau y^{i+1} + \tau(1 - \tau) [(1 - \tau)a^i + \tau b^i],$$

$$a_i = y^i - y^{i+1} + h_i f(x_i, y^i, Q), b^i = y^{i+1} - y^i - h_i f(x_{i+1}, y^{i+1}, Q),$$

где y^i и Q — приближенное решение системы (11) при заданном текущем значении параметра μ . Введем обозначения

$$\psi^i(\tau) = f[x_i + h_i \tau, \mathcal{P}_i(\tau), Q], 0 \leq \tau \leq 1,$$

$$\eta^i = \frac{y^{i+1} - y^i}{h_i} - \int_0^1 \psi^i(\tau) d\tau, \quad i = 1, 2, \dots, m - 1.$$

Тогда невязки приближенного решения системы (9) образуют составной вектор, состоящий из векторов $l, \eta^1, \eta^2, \dots, \eta^{m-1}$ и g . При этом мы будем считать, что значения интегралов в определении векторов η^i пренебрежимо мало отличаются от точных значений, если они найдены по интерполяционной формуле Котеса точности 5:

$$\int_0^1 \psi^i(\tau) d\tau \approx \frac{1}{90} \left[7\psi^i(0) + 32\psi^i\left(\frac{1}{4}\right) + 12\psi^i\left(\frac{1}{2}\right) + 32\psi^i\left(\frac{3}{4}\right) + 7\psi^i(1) \right].$$

Вычисление вектора невязок может быть полезным для косвенного контроля близости найденного решения краевой задачи и точного решения.

Приведем описание программы в соответствии с блок-схемой (см. рис. 15). Основное внимание здесь будет уделено «нестандартным» блокам, заполняемым пользователем при решении конкретной краевой задачи (1). Описание программы сопровождается примером, в качестве которого взята краевая задача (5). В связи с этим представим (5) в «стандартном» виде: $n = 2, n_0 = 1,$

$$\frac{dy_1}{dx} = f(x, y_1, y_2, Q), \quad \frac{dy_2}{dx} = f_2(x, y_1, y_2, Q), \quad 0 < x < 1, \quad (35)$$

$$l_1(y_1, y_2, Q) = 0 \text{ при } x = 0; g_1(y_1, y_2, Q) = 0 \text{ при } x = 1,$$

где $f_1 = y_2, f_2 = -Q(1 - y_1)^{-2}, l_1 = y_2, g_1 = y_1$.

1. BPR-Q (INPUT). Программа BPR-Q содержит подпрограмму INPUT, в которой помимо управляющих параметров требуется задавать сетку по x и начальное приближение решения (11) в стартовой позиции.

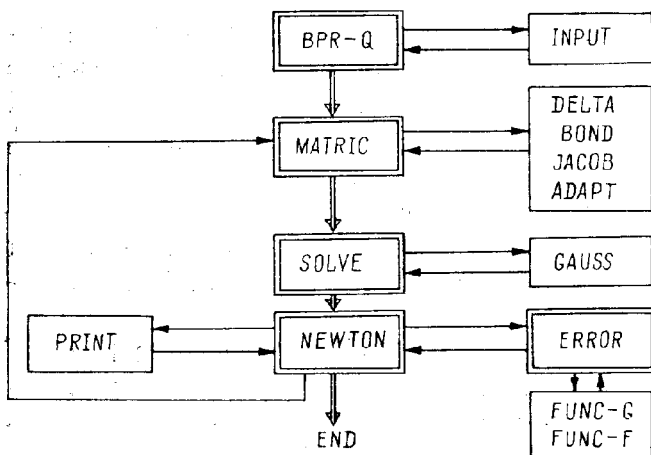


Рис. 15. Блок-схема BPR-Q.

Используем следующие обозначения:

$N \emptyset$ — параметр n , задающий размерность системы дифференциальных уравнений (1);

$N_0 \emptyset$ — параметр n_0 , определяющий число краевых условий (1) при $x = a$;

N — параметр, равный $m - 1$, т. е. числу разбиений отрезка $[a, b]$ по x ;

$D[I]$ — узел x_i сетки по $x, i = 1, 2, \dots, m$, размер массива — $D[2(N + 1)]$;

$V[I, J]$ — компонента y_j^i

вектора y^i начального приближения решения системы (11) в стартовой позиции; размеры массива — $V[N + 1, N\emptyset]$;

$H[I, J]$ — модуль эталонного шага текущего параметра системы μ , если $\mu = y_{j_*}^{i_*}$; размеры массива — $H[N + 1, N\emptyset]$;

$Q9$ — стартовое значение параметра Q ;

$H9$ — модуль эталонного шага по Q ;

$I9, J9$ — индексы i_*, j_* стартового текущего параметра μ : $1 \leq I9 \leq N + 1, 1 \leq J9 \leq N\emptyset$, если $\mu = y_{j_*}^{i_*}$; $I9 = N + 2, J9 = \emptyset$, если $\mu = Q$;

$I2$ — параметр задающий множество возможных текущих параметров системы (11):

(а) $I2 = \emptyset, \mu = y_{j_*}^{i_*}, 1 \leq i_* \leq m, 1 \leq j_* \leq m$ (т. е. $1 \leq I9 \leq N + 1, 1 \leq J9 \leq N\emptyset$); $\mu = Q, i_* = m + 1, j_* = 0$ (т. е. $I9 = N + 2, J9 = \emptyset$);

(в) $1 \leq I2 \leq N\emptyset, \mu = y_{j_*}^{i_*}, j_* = I2, 1 \leq i_* \leq m$;

(с) $I2 = N\emptyset + 1, \mu = Q, i_* = m + 1, j_* = 0$;

(d) $I2 > N\emptyset + 1, \mu = y_{j_*}^{i_*}$, где i_*, j_* сохраняют значения, присвоенные в стартовой позиции;

$H8$ — параметр ω , определяющий направление изменения μ в стартовой позиции: $H8 = 1$, если μ увеличивается, $H8 = -1$, если μ уменьшается;

$E\emptyset$ — константа сравнения ε_0 нормы невязки при итерациях по методу Ньютона — Канторовича;

$n8$ — параметр n_8 , задающий ограничение на число итераций i_0 при текущих значениях $\mu, i_0 \leq n_8, i_1 > 0$;

$n9$ — параметр n_9 , задающий ограничение на число итераций i_0 в стартовой позиции, $i_0 \leq n_9, i_1 = 0$;

$E1, E3$ — параметры $\varepsilon_1, \varepsilon_3$ ограничения текущего шага H по μ : если H_* — модуль эталонного шага по μ , то $\varepsilon_1 H_* \leq |H| \leq \varepsilon_3 H_*$;

$E2$ — параметр ε_2 «разгона» текущего шага H по μ : если H_0 — предыдущий шаг по μ , то

$$|H| = \begin{cases} \varepsilon_2 |H_0| & \text{при } \varepsilon_2 |H_0| < \varepsilon_3 H_* \\ \varepsilon_3 |H_*| & \text{при } \varepsilon_2 |H_0| \geq \varepsilon_3 H_* \end{cases}$$

j_0 — параметр j_0 , задающий номер компоненты вектор-функции $y(x)$ — решения задачи (1), применительно к которой происходит адаптация сетки (см. § 4);

n_7 — параметр n_7 , задающий номер шага i_1 по текущему параметру, на котором происходит адаптация сетки по x с последующим пересчетом задачи; после пересчета i_1 полагается равным нулю (если n_7 задать достаточно большим, то (1) будет решаться на сетке с фиксированными узлами);

d_0 — параметр d_0 , задающий ограничение на расстояния между двумя соседними узлами сетки по x (если $d_0 > b - a$, то при адаптации сетки сохраняется число узлов);

$I3$ — параметр, регулирующий обращение к программе ERROR (если $I3 = 0$, то вектор невязок приближенного решения системы (9) не вычисляется, обращение к ERROR происходит при $I3 \neq 0$, например при $I3 = 1$);

ε_4 (см. § 4), используемый для контроля направления изменения текущего параметра после адаптации сетки.

В распоряжении пользователя имеются буквы $B, C, Z, S, I, J, K, L, M$ и N для обозначений вспомогательных массивов и $B\emptyset, B1, \dots, B9, Z\emptyset, Z1, \dots, Z9, S\emptyset, S1, \dots, S9, J1, J2, \dots, J7, M1, M2, \dots, M6, K5, \dots, K8$ для обозначений простых переменных. Последние могут быть включены в список дополнительного предложения COM.

Среди других обозначений программы встречаются также следующие:

- $I\emptyset$ — счетчик итераций i_0 ;
- $I1$ — счетчик шагов i_1 по текущим параметрам;
- $K9$ — параметр k_0 (блок-схема 1);
- $I6$ — параметр i_6 (блок-схема 3).

Применительно к краевой задаче (35) имеем $N\emptyset = 2$, $M\emptyset = 1$. Отрезок $[0, 1]$ по x разобьем на 10 равных отрезков: $N = 1\emptyset$. Рассмотрим решение (35) при $Q \geq 0$. Поскольку при $Q = 0$ (т. е. $Q9 = \emptyset$) краевая задача имеет нулевое решение, то $V[I, J] = \emptyset$, $1 \leq I \leq N + 1$, $1 \leq J \leq \leq N\emptyset$. Модуль эталонного шага по Q выберем равным 0,02, а по y_j^{i*} модули эталонных шагов зададим равными 0.1: $N9 = .\emptyset 2$, $N[I, J] = .1$, $1 \leq I \leq N + 1$, $1 \leq J \leq N\emptyset$. В качестве стартового текущего параметра (11) возьмем Q . Следовательно, $I9 = 12$, $J9 = \emptyset$. Поскольку нас интересуют решения (11) при $Q > 0$, то $N8 = +1$. Положив $I2 = \emptyset$, определим множество возможных текущих параметров системы, состоящее из параметра Q и компонент вектора Y . Пусть, далее, $D2 = 1.1$, $I7 = 1$, $N7 = 1\emptyset\emptyset\emptyset$. Таким образом, решение (35) строится при фиксированном числе узлов сетки, что достигается за счет выбора $D2$, а обращение к подпрограмме DELTA исключается заданием большого числа $N7$ (первый раз решение (35) будет пересчитываться на новой сетке при $i_1 = = N7$). В данном случае адаптация сетки определяется первой компонентой вектор-функции $y(x)$ ($I7 = 1$). Полный набор значений параметров содержится в предложениях $BPR = Q$ с номерами $1\emptyset 1\emptyset - 1\emptyset 4\emptyset$ подпрограммы INPUT (см. приложение к § 6).

Заметим, что, определяя размерности массивов $D[I]$ и $V[I, J]$, следует иметь в виду возможность увеличения N при обращении к процедуре адаптации сетки, если это не исключено заданием параметров $N7$ и $D2$.

2. MATRIC. В программе MATRIC вычисляются элементы матрицы $[\Phi_Y, \Phi_Q, \Phi]$ (см. § 2) по заданному приближению решения (11) (массив $V[I, J]$ и $Q9$). Кроме того, к стандартной части MATRIC относится подпрограмма DELTA, в которой формируется новая сетка по x (см. (29)), начальное приближение решения (11), полученное в результате интерполяции решения (11) на сетке (31), и другая входная для MATRIC информация. В стартовой позиции эта информация задается в $BPR-Q$. Пользователем заполняются подпрограммы BOND, JACOB и ADAPT.

В подпрограмме BOND вычисляются элементы матрицы $[\Phi_Y, \Phi_Q, \Phi]$, связанные с краевыми условиями задачи (1). А именно, элементам массива G_{ij} размеров $n \times (n + 2)$ присваиваются, в соответствии с обозначениями (10), следующие значения:

$$G_{i1} = \frac{\partial l_i}{\partial y_1^1}, G_{i2} = \frac{\partial l_i}{\partial y_2^1}, \dots, G_{in} = \frac{\partial l_i}{\partial y_n^1}, G_{i n+1} = \frac{\partial l_i}{\partial Q}, G_{i n+2} = l_i,$$

$$i = 1, 2, \dots, n_0;$$

$$G_{n_0+i1} = \frac{\partial g_i}{\partial y_1^m}, G_{n_0+i2} = \frac{\partial g_i}{\partial y_2^m}, \dots, G_{n_0+in} = \frac{\partial g_i}{\partial y_n^m}, G_{n_0+i n+1} = \frac{\partial g_i}{\partial Q},$$

$$G_{n_0+i n+2} = g_i, i = 1, 2, \dots, n - n_0.$$

При этом учитывается, что y_j^i — элементы массива $V[I, J]$.

Применительно к (35) имеем $l_1 = y_2^1$, $g_1 = y_1^m$. Поэтому

$$\begin{bmatrix} G_{11} & G_{12} & G_{13} & G_{14} \\ G_{21} & G_{22} & G_{23} & G_{24} \end{bmatrix} = \begin{bmatrix} \frac{\partial l_1}{\partial y_1^1} & \frac{\partial l_1}{\partial y_2^1} & \frac{\partial l_1}{\partial Q} & l_1 \\ \frac{\partial g_1}{\partial y_1^m} & \frac{\partial g_1}{\partial y_2^m} & \frac{\partial g_1}{\partial Q} & g_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & y_2^1 \\ 1 & 0 & 0 & y_1^m \end{bmatrix}.$$

В подпрограмме JACOB вычисляются элементы матрицы f_y размеров $n \times n$, векторов f_Q и f размерности n (см. § 2), значения которых за-

тем присваиваются соответственно элементам массива u_{ij} , векторов R_i и Q_i . Пусть $\xi = x_k$, $q_i = y_i^k$, $f_i = f_i(\xi, q_1, q_2, \dots, q_n)$, $k = 1, 2, \dots, m$, $i = 1, 2, \dots, n$. Тогда

$$u_{i1} = \frac{\partial f_i}{\partial q_1}, u_{i2} = \frac{\partial f_i}{\partial q_2}, \dots, u_{in} = \frac{\partial f_i}{\partial q_n}, R_i = \frac{\partial f_i}{\partial Q}, O_i = f_i.$$

Применительно к (35) имеем $f_1 = q_2$, $f_2 = -Q/(1 - q_1)^2$. Следовательно, u_{ij} , R_i и Q_i , вычисляемые в подпрограмме JACOB, имеют выражения

$$\begin{aligned} \begin{bmatrix} u_{11} & u_{12} & R_1 & O_1 \\ u_{21} & u_{22} & R_2 & O_2 \end{bmatrix} &= \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \frac{\partial f_1}{\partial Q} & f_1 \\ \frac{\partial f_2}{\partial q_1} & \frac{\partial f_2}{\partial q_2} & \frac{\partial f_2}{\partial Q} & f_2 \end{bmatrix} = \\ &= \begin{bmatrix} 0 & 1 & 0 & q_2 \\ -\frac{2Q}{(1-q_1)^3} & 0 & -\frac{1}{(1-q_1)^2} & -\frac{Q}{(1-q_1)^2} \end{bmatrix}. \end{aligned}$$

В подпрограмме ADAPT, вложенной в подпрограмму DELTA, вычисляются элементы $u_{i,n+1} = \frac{\partial f_i}{\partial x}(\xi, q_1, q_2, \dots, q_n, Q)$, $i = 1, 2, \dots, n$, в связи с определением производных $\frac{d^2 y}{dx^2}$ в узлах сетки (31), которые затем используются при адаптации сетки.

Применительно к (35) имеем $u_{13} = \frac{\partial f_1}{\partial x} = 0$, $u_{23} = \frac{\partial f_2}{\partial x} = 0$. Используются следующие обозначения:

$G[I, J]$ — элемент массива G_{ij} , $i = 1, 2, \dots, n$, $j = 1, 2, \dots, n+2$; размеры массива — $G[N\emptyset, N\emptyset + 2]$;

$U[I, J]$ — элемент массива u_{ij} , $i = 1, 2, \dots, n$, $j = 1, 2, \dots, n+1$; размеры массива — $U[N\emptyset, N\emptyset + 1]$;

$Q[I]$, $R[I]$, $O[I]$ — элементы q_i , R_i и O_i соответственно, $i = 1, 2, \dots, n$; размеры массивов — $Q[2N\emptyset + 3]$, $R[N\emptyset]$, $O[N\emptyset]$;

$A[I, J]$ — элемент блочной строки матрицы Φ_Y , (для удобства обозначим $a_j = A[I, J]$, тогда матрица $[F_{y^i}^i, F_{y^{i+1}}^i]$ имеет вид

$$[F_{y^i}^i, F_{y^{i+1}}^i] = \begin{bmatrix} a_1 & \dots & a_{2n} \\ a_{2n+2} & \dots & a_{4n} \\ \vdots & \vdots & \vdots \\ a_{2n^2-2n+1} & \dots & a_{2n^2} \end{bmatrix};$$

размеры массива — $A[N+1, 2N\emptyset^2]$);

$F[I, J]$ — элемент вектора F_{Q^i} , $i = 1, 2, \dots, m-1$;

$F[M, J]$ — элемент вектора F^i , $M = I + N + 1$; размеры массива — $F[3(N+1), N\emptyset]$.

Подчеркнем, что обращение к подпрограмме BOND происходит при заданных $V[I, J]$, $I = 1, 2, \dots, N+1$, $J = 1, 2, \dots, N\emptyset$ и Q_9 . Для обращения к подпрограммам JACOB и ADAPT требуется задать массив $Q[I]$, $I = 1, 2, \dots, N$, Q_9 , а также параметр D , присвоив ему значение аргумента x . Именно в этих обозначениях определяются векторы l , f и g .

При вычислении элементов матрицы $[\Phi_Y, \Phi_Q, \Phi]$ используются вспомогательные массивы $X[N\emptyset, 4]$, $T[N\emptyset, 2N\emptyset]$, $E[N\emptyset, 2N\emptyset]$.

Отметим другие обозначения MATRIC, входящие только в предложения SOM. Пусть при $i_1 = \tilde{i}_1$ найден вектор решения системы (11) и вектор производных решения системы по параметру μ . Тогда

$Q7, Y[I, J]$ — значения Q и компоненты y_j^i решения (1), включая μ , при $i_1 = \tilde{i}_1 - 1$, $\tilde{i}_1 > 1$;

$P7, P[I, J]$ — значения производных $\frac{dQ}{d\mu}$ и $\frac{dy_j^i}{d\mu}$ при том же i_1 ;

Q8, P8, Y[M, J], P[M, J] — обозначения, аналогичные принятым при $i_1 = \tilde{i}_1$, где $M = I + N + 1$; эти же обозначения используются в стартовой позиции при $i_1 = 0$; размеры массивов — $Y[2(N+1), N\emptyset]$, $P[2(N+1), N\emptyset]$;

H5 — модуль эталонного шага текущего параметра μ ;

H6 — предыдущий шаг по μ , равный H_0 (см. (17));

H7 — текущий шаг по μ , обозначенный в (17) как H.

При использовании циклов в подпрограммах JACOB и ADAPT (а далее FUNC-F) следует учесть, что обращение к этим подпрограммам происходит в цикле, где I — счетчик. Поэтому в упомянутых подпрограммах I не может быть переменной (см. приложение к § 6).

3. SOLVE. В программе SOLVE формируется матрица $\Phi_s(S, \mu)$ и вектор $\Phi_\mu(S, \mu)$ по заданным индексам i_*, j_* текущего параметра μ (предложения SOLVE с номерами 1010—1410; см. приложение к § 6). Далее программой реализуется прямой ход прогонки решения системы алгебраических уравнений (21), (22) с матрицей $\Phi_s(S, \mu)$ согласно схемам, рассмотренным в § 4 (предложения с номерами 1420—2180). При этом используется вспомогательный массив W, к которому применяются преобразования по формулам Гаусса в подпрограмме GAUSS. Обратный ход прогонки определяет решение системы (предложения с номерами 2190—2650).

Отметим некоторые обозначения, используемые в программе:

W[I, J] — элемент вспомогательного массива W, задаваемого в программе SOLVE при обращении к подпрограмме GAUSS, при этом $W[I, 2N\emptyset + 1]$ — элемент вектора $\tilde{\Phi}_q$, $W[I, 2N\emptyset + 2]$ — элемент вектора $\tilde{\Phi}$, $W[I, 2N\emptyset + 3]$ — элемент вектора $\tilde{\Phi}_\mu$; размеры массива — $W[2N\emptyset, 2N\emptyset + 3]$;

N3, N4, L1 — параметры L_1, L_2 и L_3 соответственно; максимальное значение N3 равно $2N\emptyset + 3$, $N4 < 2N\emptyset$;

F[I, J] — элемент столбца Φ_q и $\tilde{\Phi}_q$, $N5 = N + 1$;

F[I + N5, J] — элемент столбца Φ и $\tilde{\Phi}$, $N5 = N + 1$;

F[I + N6, J] — элемент столбца Φ_μ и $\tilde{\Phi}_\mu$, $N6 = 2N5$; размеры массива — $F[3N5, N\emptyset]$.

На выходе из программы SOLVE имеем

F[I + N5, J] — значения компоненты вектора невязок с обратным знаком, соответствующей y_j^i , $1 \leq I \leq N + 1$, $1 \leq J \leq N\emptyset$;

Q[N2 + 1] — невязка для Q с обратным знаком, $N2 = 2N\emptyset$;

F[I + N6, J] — соответствующее значение компоненты вектора производных решения по параметру μ с обратным знаком (если $K9 = 2$);

Q[N2 + 2] — производная $\frac{dQ}{d\mu}$ с обратным знаком.

4. NEWTON. В программе NEWTON формируется приближение решения системы (11) на очередной итерации — массив V[I, J], $1 \leq I \leq N5$, $1 \leq J \leq N\emptyset$, и Q9 (предложения NEWTON с номерами 1090—1099; см. приложение к § 6). Одновременно здесь вычисляется норма невязки — константа C9.

Массив F[I + N6, J], $1 \leq I \leq N5$, $1 \leq J \leq N\emptyset$, и элемент Q[N2 + 2] содержат значения компонент вектора производных решения системы по текущему параметру μ (см. обозначения п. 3), которые вычисляются в программе SOLVE после сходимости итерационного процесса ($K9 = 2$). Эти значения используются для отыскания нового текущего параметра системы (т. е. индексов I9, J9), отвечающего тому же значению I1, и нового вектора производных решения по найденному текущему параметру через процедуру нормировки (предложения с номерами 1470—1820).

В стартовой позиции ($I1 = \emptyset$) определяется шаг H7 по текущему параметру, а значения компонент векторов решения системы и производных решения системы по текущему параметру (см. обозначения п. 2)

присваиваются элементам массива $Y[I + N5, J]$ и $Q8$, элементам массива $P[I + N5, J]$ и $P8$, $1 \leq I \leq N5$, $1 \leq J \leq N\emptyset$, соответственно (предложения с номерами 187 \emptyset — 2 \emptyset 5 \emptyset). Далее строится начальное приближение решения системы по формуле (13), отвечающее значению $I1 = 1$ (предложения с номерами 129 \emptyset — 134 \emptyset).

Если $I1 > \emptyset$, то описание случая $I1 = \emptyset$ дополняется запоминанием результатов предыдущего шага с номером $I1 - 1$, используя для этого массив $Y[I, J]$ и $Q7$, массив $P[I, J]$ и $P7$, $1 \leq I \leq N5$, $1 \leq J \leq N\emptyset$ (предложения с номерами 2 \emptyset 7 \emptyset — 229 \emptyset). Начальное приближение решения системы, отвечающее шагу с номером $I1 + 1$, определяется по формулам (17) (предложения с номерами 136 \emptyset — 145 \emptyset).

Напомним, что к формулам (13) и (17) мы обращаемся также при делении шага $N7$ пополам, что связано с ограничением на число итераций (предложения с номерами 117 \emptyset — 126 \emptyset). Кроме того, в результате адаптации сетки ($I1 \geq N7$) мы возвращаемся к построению начального приближения, положив $I1 = \emptyset$, и сопровождаем его контролем правильности выбора направления изменения текущего параметра (предложения с номерами 234 \emptyset — 215 \emptyset).

Подпрограмма PRINT, заполняемая пользователем, предназначена для выдачи результатов на печать и задания признака окончания работы программы. При обращении к PRINT в распоряжении пользователя имеются, в частности, следующие значения переменных:

$D[I]$ — сетка по x , $1 \leq I \leq N5$,
 $V[I, J]$, $Q9$ — решение системы (2), т. е. значения $y_j^i = y_i(x_j)$ и Q ,
 $1 \leq I \leq N5$, $1 \leq J \leq N\emptyset$,
 $F[I + N6, J]$, $Q[N2 + 2]$ — производные по текущему параметру $\frac{dy_j^i}{d\mu}$ и $\frac{dQ}{d\mu}$, $1 \leq I \leq N5$, $1 \leq J \leq N\emptyset$,
 $I9, J9$ — индексы μ ,

а также другие переменные, содержащиеся в предложениях COM.

Если $I3 = 1$ (или $I3 \neq \emptyset$), то на печать могут быть выданы значения элементов массива $F[I, J]$, $1 \leq I \leq N5$, $1 \leq J \leq N\emptyset$, — компоненты вектора невязок системы (9) (см. п. 5).

В связи с рассматриваемой в качестве примера задачей (35) на печать выдается таблица значений: $Y(\emptyset)$, соответствующее $y_1(\emptyset)$, значение $Q9$, отклонение $Q9$ от точного значения, индексы $I9$ и $J9$ текущего параметра μ .

5. ERROR. В программе ERROR вычисляются невязки системы (9), значения которых присваиваются элементам массива $F[I, J]$, $1 \leq I \leq N5$, $1 \leq J \leq N\emptyset$ (размеры массива — $F[3N5, N\emptyset]$ — были определены ранее). При этом $F[I, J]$ — невязка уравнения (9) с номером $N\emptyset(I - 1) + J$.

Напомним, что для обращения к ERROR требуется задать $I3 \neq 0$. В противном случае, т. е. при $I3 = \emptyset$, невязки не вычисляются. Фактически программа ERROR служит подпрограммой NEWTON, обращение к которой содержится в предложении с номером 184 \emptyset (см. приложение к § 6). По завершению работы ERROR управление передается на предложение NEWTON с номером 185 \emptyset .

Для вычисления невязок пользователем заполняются подпрограммы FUNC-G и FUNC-F, частично повторяющие BOND и JACOB.

В подпрограмме FUNC-G вычисляются элементы массива $G[I, N\emptyset + 2]$, $1 \leq I \leq N\emptyset$ (см. § 6, п. 2, подпрограмма BOND). Обращение к FUNC-G совпадает с обращением к BOND. Применительно к рассматриваемому примеру (35) имеем $G_{14} = y_2^1$, $G_{24} = y_1^m$.

В подпрограмме FUNC-F вычисляются элементы массива $O[I]$, $1 \leq I \leq N\emptyset$ (см. подпрограмму JACOB, п. 2). Обращение к FUNC-F совпадает с обращением к JACOB. В рассматриваемом примере $O_1 = q_2$, $O_2 = -Q/(1 - q_1)^2$.

**§ 7. ОБ ИСПОЛЬЗОВАНИИ УПРАВЛЯЮЩИХ ПАРАМЕТРОВ
ПРОГРАММЫ BPR-Q**

1. Вначале обратим внимание на выбор параметра I2, задание которого влияет на процедуру параметризации (см. обозначения в § 6, п. 1). При заполнении нестандартных блоков по-прежнему имеется в виду краевая задача (35), где в стартовой позиции в качестве текущего выбран параметр Q, равный нулю: Q9 = ∅, I9 = N + 1, J9 = ∅.

Пусть I2 = ∅, т. е. отсутствует ограничение на множество возможных текущих параметров системы. Результаты вычислений отражены на рис. 16 в виде таблицы. Напомним, что в соответствии с заполнением INPUT решение (35) строится на сетке по x с фиксированным числом узлов при равномерном разбиении. Значение I9 совпадает с номером i узла сетки x_i, а J9 с номером компоненты вектор-функции y(x) (35). В пояснении к таблице приводится формулировка (35) в виде краевой задачи для дифференциального уравнения второго порядка относительно неизвестной функции y(x):

$$\frac{d^2y}{dx^2} = -\frac{Q}{(1-y)^2}, \quad 0 < x < 1, \quad (36)$$

$$\frac{dy}{dx} = 0 \text{ при } x = 0, \quad y = 0 \text{ при } x = 1.$$

Поэтому печать результатов, где, например, I9 = 6, J9 = 2, означает, что

BPR-Q

TWO POINT BOUNDARY VALUE PROBLEM :

0 < X < 1

Y''(X) + Q/[1-Y(X)]^2 = 0

Y'(0) = 0 , Y(1) = 0

START : Q=0, I9=N+2, J9=0

I2=0, N=10, N7=10000

Y(0)	Q	ERR(Q)	I9	J9
0.00000	0.00000	-1.366E-14	11	2
0.05087	0.09329	-3.129E-07	11	2
0.10366	0.17294	-6.557E-07	11	2
0.15868	0.23859	-9.239E-07	11	2
0.21635	0.28977	-1.013E-06	11	2
0.27724	0.32587	-1.073E-06	11	2
0.34219	0.34602	-1.073E-06	11	2
0.41248	0.34897	-8.345E-07	10	2
0.48864	0.33319	-6.557E-07	8	2
0.56870	0.29918	-5.960E-07	6	2
0.64761	0.25227	-4.172E-07	5	2
0.72191	0.19977	-1.043E-06	4	2
0.78711	0.14999	-3.755E-06	3	2
0.83876	0.10993	-1.164E-05	3	2
0.88159	0.07740	-3.411E-05	2	2
0.91160	0.05550	-7.336E-05	2	2
0.93445	0.03957	-1.048E-04	2	2
0.95327	0.02719	-4.604E-05	2	2
0.97009	0.01699	+2.530E-04	2	2
0.98646	0.00720	-1.127E-05	2	2
0.99084	0.00405	-7.424E-04	2	2
0.99683	0.00058	-1.031E-03	2	2

DONE

Рис. 16.

TWO POINT BOUNDARY VALUE PROBLEM :

$$0 < x < 1$$

$$Y''(x) + Q/[1 - Y(x)] + 2 = 0$$

$$Y'(0) = 0, \quad Y(1) = 0$$

START : Q=0, I9=N+2, J9=0
I2=1, N=10, N7=10000

Y(0)	Q	ERR(Q)	I9	J9
0.00000	0.00000	-1.366E-14	1	1
0.05000	0.09183	-2.831E-07	1	1
0.10000	0.16796	-7.451E-07	1	1
0.15000	0.22936	-8.643E-07	1	1
0.20000	0.27701	-9.537E-07	1	1
0.25000	0.31189	-1.073E-06	1	1
0.30000	0.33498	-1.132E-06	1	1
0.35000	0.34727	-1.013E-06	1	1
0.40000	0.34976	-1.013E-06	1	1
0.45000	0.34345	-6.557E-07	1	1
0.50000	0.32936	-5.960E-07	1	1
0.55000	0.30850	-4.768E-07	1	1
0.60000	0.28193	-4.172E-07	1	1
0.65000	0.25069	-4.172E-07	1	1
0.70000	0.21587	-7.749E-07	1	1
0.75000	0.17858	-1.848E-06	1	1
0.80000	0.13998	-4.917E-06	1	1
0.85000	0.10130	-1.523E-05	1	1
0.90000	0.06385	-5.522E-05	1	1
0.95000	0.02928	-7.013E-05	1	1

□ ONE

Рис. 17.

текущим параметром является $\frac{dy}{dx}(x_0)$. Таблица отражает изменение индексов текущего параметра в соответствии с изменением $y(0)$. Отметим, что всякий раз $J9 = 2$, а $I9$ убывает с ростом $y(0)$ (значение $y(0)$ является также результатом вычислений). График зависимости Q от λ , $\lambda = y(0)$, приведен на рис. 2.

Задав $I2 = 1$, мы ограничим множество возможных текущих параметров системы, в качестве которых теперь могут выбираться лишь $y(x_i)$ (в этом случае $J9 = I2 = 1$). Результаты вычислений приведены на рис. 17. Мы видим, что текущим параметром системы всегда оказывался $y(0)$.

Наконец, пусть $I2 = 3$ (т. е. $I2 = N\emptyset + 1$), что означает фиксирование Q в качестве текущего параметра. Результаты вычислений, представленные на рис. 18, отражают процесс уменьшения шага по Q по мере приближения к точке ветвления. При этом достаточно точно определено значение Q , равное 0.35, при котором происходит ветвление решений (36). Программа прекратила работу, как только шаг по Q оказался меньше заданного значения.

2. Информация об индексах текущих параметров позволяет в ряде случаев, как и в рассматриваемом примере, задавать не слишком точное начальное приближение за счет правильного выбора текущего параметра задачи в стартовой позиции.

На рис. 19 представлено решение (36) при фиксированном значении $y(0)$, $y(0) = 0.8$. Таким образом, $y(0)$ — стартовый текущий параметр задачи с индексами $I9 = 1$, $J9 = 1$. В качестве начального приближения

TWO POINT BOUNDARY VALUE PROBLEM :

$$0 < x < 1$$

$$Y''(x) + Q/[1 - Y(x)] + 2 = 0$$

$$Y'(0) = 0, \quad Y(1) = 0$$

START : Q=0, I9=N+2, J9=0

I2=3, N=10, N7=10000

Y(0)	Q	ERR(Q)	I9	J9
0.00000	0.00000	-1.366E-14	12	0
0.01017	0.02000	-6.706E-08	12	0
0.02071	0.04000	-8.941E-08	12	0
0.03165	0.06000	-2.161E-07	12	0
0.04304	0.08000	-3.278E-07	12	0
0.05492	0.10000	-3.576E-07	12	0
0.06737	0.12000	-5.066E-07	12	0
0.08045	0.14000	-4.470E-07	12	0
0.09427	0.16000	-6.557E-07	12	0
0.10895	0.18000	-7.451E-07	12	0
0.12464	0.20000	-7.451E-07	12	0
0.14157	0.22000	-9.239E-07	12	0
0.16004	0.24000	-8.047E-07	12	0
0.18050	0.26000	-9.537E-07	12	0
0.20368	0.28000	-1.073E-06	12	0
0.23088	0.30000	-1.013E-06	12	0
0.26498	0.32000	-1.073E-06	12	0
0.31586	0.34000	-1.192E-06	12	0
0.33670	0.34500	-9.537E-07	12	0
0.35162	0.34750	-8.941E-07	12	0
0.36226	0.34875	-1.073E-06	12	0
0.36983	0.34938	-8.941E-07	12	0
0.37519	0.34969	-1.013E-06	12	0
0.37899	0.34984	-8.941E-07	12	0
0.38166	0.34992	-8.941E-07	12	0
0.38704	0.35000	-8.941E-07	12	0

I1= 26 ABS(N7) < E1*N5, STOP

DONE

Рис. 18.

решения используется функция $y^0(x) = y(0)(1 - x^2)$. Результаты вычислений представлены на рис. 19. Если, например, в этой ситуации взять соответствующее значение Q в качестве стартового текущего параметра задачи, то итерационный процесс окажется расходящимся. Одновременно этот пример иллюстрирует проблему уточнения решения, полученного на первоначально заданной сетке за счет адаптации сетки. В таблицах на рис. 19 строка содержит значения узлов сетки по x , $y(x)$, $\frac{dy}{dx}(x)$ и невязки решения системы (9), найденные путем обращения к программе ERROR. Таблицам предшествуют печать значений $S9$ (нормы невязок на итерациях), вычисленное значение Q и погрешность вычисления Q .

Тексты подпрограмм INPUT и PRINT, отвечающие рис. 19, приведены в приложении к § 7. Обращаем внимание на то, что в число управляющих параметров здесь включен I6, I6 = -1 и, следовательно, в строке с номером 2000 из BPR-Q задаются только I0 и I1. Для печати S9 программа NEWTON дополняется предложением: 1105 PRINT «S9 =»; S9.

TWO POINT BOUNDARY VALUE PROBLEM :

$$0 < X < 1$$

$$Y''(X) + Q/[1 - Y(X)]^2 = 0$$

$$Y'(0) = 0, \quad Y(1) = 0$$

START : Q = .15, I9 = 1, J9 = 1, Y(X) = .8*(1-X^2)
I2 = 4, N = 10, N7 = 0, D2 = 1.1, I3 = 1, I6 = -1

C9 = 1
C9 = 3.46788E-02
C9 = 3.45021E-04
C9 = 1.01286E-07
C9 = 6.98802E-08

Q = .139981 ERR(Q) = -4.91738E-06

X	Y(X)	Y'(X)	ERR(Y)	ERR(Y')
0.00000	0.80000	+0.00000	+0.000E+00	-4.768E-07
0.10000	0.78299	-0.33132	-3.076E-04	-3.576E-07
0.20000	0.73661	-0.58051	+3.304E-04	-1.550E-06
0.30000	0.66982	-0.74296	+9.584E-05	-2.027E-06
0.40000	0.58996	-0.84684	-1.025E-05	-9.537E-07
0.50000	0.50161	-0.91551	-1.812E-05	-2.027E-06
0.60000	0.40755	-0.96299	-9.894E-06	-2.027E-06
0.70000	0.30945	-0.99724	-5.305E-06	-1.907E-06
0.80000	0.20839	-1.02286	-3.695E-06	-2.146E-06
0.90000	0.10507	-1.04263	-1.252E-06	-1.907E-06
1.00000	0.00000	-1.05827	-8.643E-07	+0.000E+00

C9 = 2.27768E-04
C9 = 1.31321E-07
C9 = 1.16932E-07

Q = .139986 ERR(Q) = 2.98023E-07

X	Y(X)	Y'(X)	ERR(Y)	ERR(Y')
0.00000	0.80000	+0.00000	+0.000E+00	+8.494E-07
0.05659	0.79445	-0.19446	-5.627E-05	-1.907E-06
0.12923	0.77205	-0.41432	+3.672E-05	-1.073E-06
0.23235	0.71681	-0.64128	+3.119E-04	-3.576E-07
0.30681	0.66473	-0.75155	+1.502E-05	-2.146E-06
0.38667	0.60116	-0.83541	-6.199E-06	-1.431E-06
0.47617	0.52326	-0.90145	-1.192E-05	-1.669E-06
0.57796	0.42868	-0.95385	-1.299E-05	-1.550E-06
0.69556	0.31387	-0.99591	-1.055E-05	-2.146E-06
0.83598	0.17144	-1.03052	-1.052E-05	-1.907E-06
1.00000	0.00000	-1.05826	-7.421E-06	+0.000E+00

DONE

Рис. 19.

3. Пусть о краевой задаче (1) известна информация, например, типа той, что представлена на рис. 17. Предположим, что этого достаточно для задания приемлемого начального приближения решения (4) при некотором фиксированном значении текущего параметра и для того, чтобы восстановить само решение. Используя затем полученное в стартовой позиции решение, найдем решение (1) при заданном значении Q , равном

BPR-Q

TWO POINT BOUNDARY VALUE PROBLEM :

$$0 < X < 1$$

$$Y''(X) + Q/[1 - Y(X)] + 2 = 0$$

$$Y'(0) = 0, \quad Y(1) = 0$$

START : N=10, I9=1, J9=1, I2=0

$$Y_0 = .5, \quad Y(X) = Y_0 * (1 - X^2)$$

Y(0) = 0.50000	Q = 0.32936	Q' = 0.27897
Y(0) = 0.58160	Q = 0.29231	Q' = 0.43937
Y(0) = 0.66190	Q = 0.24268	Q' = 0.52600
Y(0) = 0.73732	Q = 0.18821	Q' = 0.51981
Y(0) = 0.72160	Q = 0.20000	Q' = 0.52747

X	Y(X)	Y'(X)
0.00000	0.72160	+0.00000
0.10000	0.70889	-0.25045
0.20000	0.67282	-0.46287
0.30000	0.61805	-0.62413
0.40000	0.54948	-0.74090
0.50000	0.47096	-0.82506
0.60000	0.38522	-0.88666
0.70000	0.29414	-0.93280
0.80000	0.19901	-0.96820
0.90000	0.10075	-0.99598
1.00000	-0.00000	-1.01823

DONE

Рис. 20.

Q_0 . С этой целью в подпрограмму PRINT включается подпрограмма SOLV-Q. Для обращения к SOLV-Q требуется задать Q_0 (обозначение $Q\emptyset$) и относительную погрешность ε_5 (обозначение E5) приближения Q к Q_0 . Кроме того, выбором I2 в стартовой позиции (подпрограмма INPUT) исключается Q из возможных текущих параметров, $0 \leq I2 \leq N\emptyset$. (Если из информации о краевой задаче (1) следует, что возможно продолжение решения по параметру Q , то обращение к подпрограмме SOLV-Q становится излишним.) В предлагаемом варианте программа BPR-Q завершает работу определением решения при заданном значении Q .

Пусть μ — текущий параметр на i_1 -м шаге, равный μ_2 , μ_1 — значение μ на $(i_1 - 1)$ -м шаге, $i_1 > 0$, и пусть известны значения $Q(\mu)$ и $\frac{dQ}{d\mu}(\mu)$ при $\mu = \mu_1$ и $\mu = \mu_2$, причем

$$\text{SGN}[Q(\mu_1) - Q_0] \neq \text{SGN}[Q(\mu_2) - Q_0].$$

Тогда, используя приближенное представление $Q(\mu)$ на отрезке $[\mu_1, \mu_2]$ в виде кубической параболы, мы можем определить $\mu = \mu_0$, $\mu_0 \in (\mu_1, \mu_2)$, при котором значение параболы равно Q_0 . В дальнейшем, на $(i_1 + 1)$ -м шаге, либо вновь определяется μ_0 точно таким же образом, если выполнено условие на знак $Q(\mu_1) - Q_0$, либо шаг по текущему параметру сохраняет предшествующее значение.

Текст SOLV-Q содержится в подпрограмме PRINT, где формируется печать результатов для примера (36) (см. приложение к § 7). Процесс продолжения решения отражен на рис. 20. При завершении работы программы на печать выдаются результаты при $Q = 0.2$.

TWO POINT BOUNDARY VALUE PROBLEM :

$$0 < X < 1$$

$$Y''(X) + Q/[1 - Y(X)] + 2 = 0$$

$$Y'(0) = 0, \quad Y(1) = 0$$

START : N=10, I9=1, J9=1, I2=0

Q=0, Y(X)=0

Y(0) = 0.00000	Q = 0.00000	Q' = -1.00000
Y(0) = 0.05087	Q = 0.09329	Q' = -0.86524
Y(0) = 0.10366	Q = 0.17294	Q' = -0.72720
Y(0) = 0.15868	Q = 0.23859	Q' = -0.58505
Y(0) = 0.21635	Q = 0.28977	Q' = -0.43758
Y(0) = 0.27724	Q = 0.32587	Q' = -0.28294
Y(0) = 0.34219	Q = 0.34602	Q' = -0.11809
Y(0) = 0.41248	Q = 0.34897	Q' = +0.06226
Y(0) = 0.38833	Q = 0.35000	Q' = -0.00004

X	Y(X)	Y'(X)
0.00000	0.38833	+0.00000
0.10000	0.38367	-0.09308
0.20000	0.36981	-0.18341
0.30000	0.34715	-0.26867
0.40000	0.31630	-0.34724
0.50000	0.27796	-0.41826
0.60000	0.23291	-0.48154
0.70000	0.18190	-0.53737
0.80000	0.12565	-0.58635
0.90000	0.06483	-0.62919
1.00000	0.00000	-0.66664

DONE

Рис. 21.

4. Пусть на i_1 -м шаге известны $Q(\mu)$ и $\frac{dQ}{d\mu}(\mu)$ при $\mu = \mu_2$ и $\mu = \mu_1$, причем

$$\text{SGN} \left[\frac{dQ}{d\mu}(\mu_1) \right] \neq \text{SGN} \left[\frac{dQ}{d\mu}(\mu_2) \right].$$

Тогда приближение $Q(\mu)$ на отрезке $[\mu_1, \mu_2]$ в виде кубической параболы позволяет найти $\mu = \mu_0$, $\mu_0 \in (\mu_1, \mu_2)$, при котором значение производной кубической параболы по μ равно нулю. В дальнейшем, на $(i_1 + 1)$ -м шаге, либо вновь определяется μ_0 точно таким же способом, если выполнено условие на знак $\frac{dQ}{d\mu}(\mu_1)$, либо шаг по текущему параметру сохраняет предшествующее значение.

Для определения значения параметра Q , при котором происходит ветвление решений (1) по параметру Q , используется подпрограмма EXTREM, включенная в подпрограмму PRINT (см. приложение к § 7). Для обращения к EXTREM требуется задать ϵ_5 (обозначение E5), так что если $\left| \frac{dQ}{d\mu}(\mu) \right| < \epsilon_5$, то проблема считается решенной. Кроме того, в подпрограмме INPUT задается I2, подчиненное условию $0 \leq I2 \leq N0$. Завершается работа программы выдачей на печать результатов при $\frac{dQ}{d\mu} = 0$ (см. рис. 21).

§ 8. ЧИСЛЕННЫЕ ПРИМЕРЫ

Предлагаемые здесь примеры иллюстрируют возможности программы BPR-Q. С другой стороны, для читателей, пожелавших реализовать программу, эти примеры могут служить тестами при отладке программы и пособием по ее использованию. С этой целью результаты вычислений сопровождаются текстами нестандартных подпрограмм, заполняемых пользователем.

Пример 1. Рассмотрим краевую задачу, модулирующую экзотермическую реакцию первого порядка в зерне катализатора:

$$\begin{aligned}
 0 < x < 1, \quad C = 30, \quad B = 1/60, \quad Q \geq 0, \\
 \frac{dy_1}{dx} &= y_2, \\
 \frac{dy_2}{dx} &= -\frac{2}{x} y_2 - Q(C - y_1) \exp\left(\frac{y_1}{1 + By_1}\right), \\
 y_2 &= 0 \text{ при } x = 0, \quad y_1 = 0 \text{ при } x = 1.
 \end{aligned} \tag{37}$$

Здесь x — нормированное расстояние от центра зерна, $y_1(x)$ — нормированная температура. Этот пример интересен тем, в частности, что в процессе продолжения решения по параметру область больших градиентов решения перемещается из окрестности $x = 0$ к окрестности $x = 1$, и поэтому использование процедуры адаптации сетки носит в данном случае принципиальный характер. Столь же существенна и параметризация: число решений (37) меняется в зависимости от Q и доходит до 7 в окрестности $\sqrt{Q} = 0.257$.

Для наглядности представления множественности решений (37) вводится так называемый фактор эффективности

$$\eta = \frac{3}{\varphi[y_1(1)]} \int_0^1 x^2 \varphi[y_1(x)] dx,$$

где $\varphi(u) = (C - u) \exp\left(\frac{u}{1 + Bu}\right)$. В силу (37) имеет место равенство

$$Q \int_0^1 x^2 \varphi[y_1(x)] dx = -y_2(1),$$

из которого следует, что $\eta = -\frac{3}{QC} y_2(1)$. Множественность решений определяется после построения зависимости $\eta = \eta(Q)$ на решениях (37).

Рассмотрим содержание нестандартных подпрограмм BPR-Q в связи с (37). Как и в примере (35), здесь $n = 2$, $n_0 = 1$. Краевые условия (35) и (37) совпадают. Поэтому содержание подпрограммы BOND остается таким же, как и в случае (35). Применительно к (37) имеем (см. § 6, п. 2) $f_1 = q_2$, $f_2 = -\frac{2}{\xi} q_2 - Q\varphi(q_1)$. Следовательно, в подпрограмме JASOB вычисляются элементы матрицы

$$\begin{bmatrix} u_{11} & u_{12} & R_1 & O_1 \\ u_{21} & u_{22} & R_2 & O_2 \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \frac{\partial f_1}{\partial Q} & f_1 \\ \frac{\partial f_2}{\partial q_1} & \frac{\partial f_2}{\partial q_2} & \frac{\partial f_2}{\partial Q} & f_2 \end{bmatrix},$$

причем $u_{11} = 0$, $u_{12} = 1$, $R_1 = 0$, $O_1 = q_2$, $u_{21} = -Q \frac{d\varphi}{dq_1}$, $u_{22} = -\frac{2}{\xi}$, $R_2 = -\varphi(q_1)$, $O_2 = -\frac{2}{\xi} q_2 - Q\varphi(q_1)$, если $\xi > 0$ и $u_{21} = -\frac{Q}{3} \frac{d\varphi}{dq_1}(q_1)$, $u_{22} =$

TWO POINT BOUNDARY VALUE PROBLEM :

$$0 < X < 1, \quad C=30, \quad B=1/60$$

$$Y'' + 2*Y'/X + Q*(C-Y)*EXP[Y/(1+B*Y)] = 0$$

$$Y'(0) = 0, \quad Y(1) = 0$$

Y(0)	SQR(Q)	LOG(FACTOR)	Y'(1)	Q'
0.0	0.00000	0.00000	+0.00000	+1.99999E-01
0.5	0.26797	0.08176	-0.86686	+9.70950E-02
1.0	0.32347	0.15898	-1.50887	+3.98737E-02
1.5	0.34062	0.23144	-1.97685	+8.93096E-03
1.5	0.34062	0.23144	-1.97685	+8.93062E-03
2.0	0.34065	0.29892	-2.30962	-6.97727E-03
2.5	0.33233	0.36117	-2.53702	-1.43739E-02
3.0	0.32008	0.41797	-2.68216	-1.70403E-02
3.0	0.32008	0.41797	-2.68214	-1.70416E-02
3.5	0.30631	0.46908	-2.76309	-1.71617E-02
4.0	0.29241	0.51427	-2.79407	-1.59823E-02
4.5	0.27918	0.55330	-2.78644	-1.42048E-02
4.5	0.27918	0.55329	-2.78654	-1.42052E-02
5.0	0.26708	0.58596	-2.74949	-1.22200E-02
5.5	0.25637	0.61210	-2.69048	-1.02244E-02
6.0	0.24717	0.63159	-2.61575	-8.31315E-03
6.0	0.24716	0.63162	-2.61556	-8.31582E-03
6.5	0.23955	0.64447	-2.53074	-6.52529E-03
7.0	0.23354	0.65069	-2.44012	-4.86103E-03
7.5	0.22913	0.65048	-2.34775	-3.31382E-03
7.5	0.22912	0.65046	-2.34742	-3.33947E-03
8.0	0.22625	0.64437	-2.25709	-1.90327E-03
8.5	0.22490	0.63292	-2.17209	-5.58624E-04
9.0	0.22498	0.61698	-2.09540	+6.94989E-04
9.0	0.22496	0.61703	-2.09521	+6.42903E-04
9.5	0.22630	0.59793	-2.02913	+1.76739E-03
10.0	0.22881	0.57679	-1.97572	+2.76480E-03
10.5	0.23227	0.55497	-1.93627	+3.59867E-03
10.5	0.23228	0.55483	-1.93579	+3.57120E-03
11.0	0.23643	0.53363	-1.91004	+4.17397E-03
11.5	0.24102	0.51423	-1.89813	+4.54066E-03
12.0	0.24576	0.49749	-1.89902	+4.65916E-03
12.0	0.24578	0.49734	-1.89860	+4.64995E-03
12.5	0.25042	0.48362	-1.90967	+4.52087E-03
13.0	0.25474	0.47321	-1.92938	+4.18615E-03
13.5	0.25859	0.46604	-1.95553	+3.69027E-03
13.5	0.25861	0.46590	-1.95517	+3.72301E-03
14.0	0.26191	0.46147	-1.98503	+3.13763E-03
14.5	0.26459	0.45959	-2.01713	+2.49968E-03
15.0	0.26664	0.45985	-2.04970	+1.84992E-03
15.0	0.26669	0.45968	-2.04980	+1.89522E-03
15.5	0.26818	0.46155	-2.08166	+1.29351E-03

DONE

Рис. 22.

$= 0, R_2 = -\frac{1}{3} \varphi(q_1), O_2 = -\frac{1}{3} Q \varphi(q_1),$ если $\xi = 0.$ Здесь

$$\frac{d\varphi}{dq_1}(q_1) = \left[\frac{C - q_1}{(1 + Bq_1)^2} - 1 \right] \exp\left(\frac{q_1}{1 + Bq_1}\right).$$

В подпрограмме ADAPT вычисляются u_{13} и u_{23} :

$$u_{13} = \frac{\partial f_1}{\partial x} = 0, \quad u_{23} = \frac{\partial f_2}{\partial x} = \frac{2}{\xi^2} q_2,$$

если $\xi > 0$, и $u_{13} = u_{23} = 0$, если $\xi < 0$.

Наконец, в подпрограммах FUNC-G и FUNC-F вычисляются G_{14} , G_{24} (FUNC-G) и O_1 , O_2 (FUNC-F).

В подпрограмме INPUT приведены значения параметров программы с целью построения таблицы, изображенной на рис. 22. Формирование печати таблицы содержится в подпрограмме PRINT. Тексты перечисленных подпрограмм приведены в приложении к § 8.

Таблица на рис. 22 содержит результаты вычислений, отвечающие стартовой позиции, определенной в подпрограмме INPUT. Отметим, что поскольку $I_2 = 1$, то в качестве текущего параметра μ выбирается y_1^i , т. е. $y_1(x_i)$, $i = 1, 2, \dots, m$. Кроме $\log(\eta)$ (в таблице LOG(FACT)) печатаются соответствующие значения $\frac{dQ}{d\mu}$ (в таблице Q'). Таблица иллюстрирует вариант выдачи информации, необходимой при численном исследовании множественности решений краевой задачи. Для этого, опираясь на табличные данные, требуется построить график, например, зависимости $\log(\eta)$ от \sqrt{Q} , $Q > 0$. При этом значения $y_1(0)$ (в таблице $Y(\emptyset)$) меняются от 0 до C , $C = 30$. В таблице приведены данные, соответствующие изменению $y_1(0)$ от 0 до 15. Мы искусственно ограничились этими значениями $y_1(0)$, приняв неравенство $y_1(0) > 15$ за условие прекращения работы программы (предложение PRINT с номером 3Ø8Ø). Обратим внимание на то, что табличным данным отвечает выбор только y_1^1 . Повторение в таблице значений $Y(\emptyset)$ является результатом обращения к подпрограмме DELTA ($N_7 = 3$, см. INPUT).

Более полно результаты численного исследования множественности решений (37) представлены на рис. 23. Как уже отмечалось, при $\sqrt{Q} = 0.257$ краевая задача имеет семь решений, каждое из которых характеризуется своим значением η . На рис. 23 эти значения обозначены цифрами 1 ÷ 7. На рис. 24 приведены соответствующие графики семи решений, а именно, графики функции $1 + BY(x)$, $Y(x) = y_1(x)$, $B = 1/60$. Кроме того, здесь введены дополнительные шкалы по x для иллюстрации поведения 3 ÷ 7-го решений в области больших градиентов.

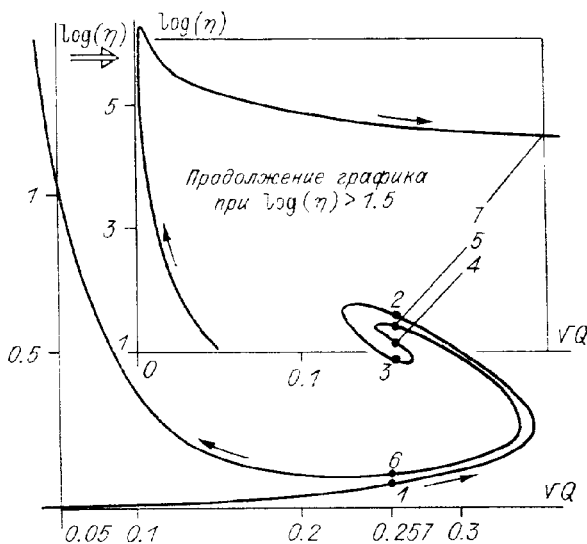


Рис. 23.

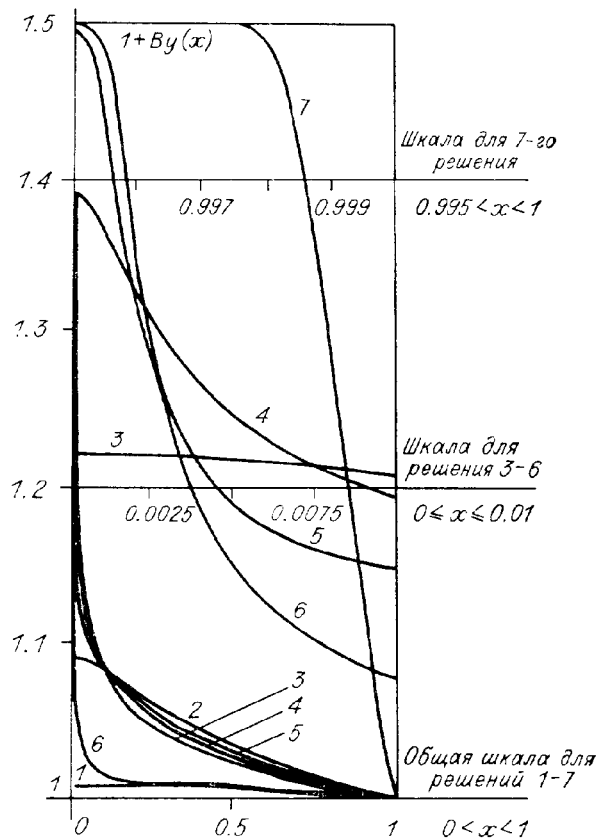


Рис. 24.

Пример 2. Рассмотрим еще один пример, относящийся к математическому моделированию стационарных режимов каталитических реакций. При описании реактора с кипящим слоем возникает следующая краевая задача:

$$0 < x < 1,$$

$$\begin{aligned} \frac{dy_1}{dx} &= y_2, \\ \frac{dy_2}{dx} &= s_0 [(s_1 + s_2)y_1 - s_2y_3 - Q\varphi(y_1)y_4], \\ \frac{dy_3}{dx} &= s_2(y_1 - y_3), \\ \frac{dy_4}{dx} &= -s_3\varphi(y_1)y_4, \\ \varphi(y_1) &= \frac{1}{s_5 + \exp\left[-\frac{1+s_4}{1+s_6(y_1+s_4)}\right]}, \end{aligned} \quad (38)$$

$$y_2 = y_3 = 0, \quad y_4 = 1 \text{ при } x = 0, \quad y_2 = 0 \text{ при } x = 1.$$

Здесь функции $y_1(x)$, $y_3(x)$ и $y_4(x)$ описывают соответственно безразмерные распределения температуры твердой фазы, температуры газообразной фазы и концентрации. Параметры задачи имеют значения $s_0 = 10$, $s_1 = 8$, $s_2 = s_3 = 5$, $s_4 = -4$, $s_5 = 1/2$, $s_6 = 1/20$.

Очевидно, в подпрограмме INPUT в данном случае требуется задать $n = 4$, $n_0 = 3$. Значения других управляющих параметров, начальное приближение решения (38) при $Q = 0$ содержатся в тексте INPUT (тексты INPUT, BOND и других нестандартных подпрограмм см. в приложении к § 8).

Краевые условия задачи (38) в обозначениях § 6 имеют вид

$$l_1 = y_2^1, \quad l_2 = y_3^1, \quad l_3 = y_4^1 - 1, \quad g_1 = y_2^m.$$

Отсюда следует (см. § 6 п. 2) такое правило задания массива G_{ij} в подпрограмме BOND:

$$\begin{bmatrix} G_{11} & G_{12} & G_{13} & G_{14} & G_{15} \\ G_{21} & G_{22} & G_{23} & G_{24} & G_{25} \\ G_{31} & G_{32} & G_{33} & G_{34} & G_{35} \\ G_{41} & G_{42} & G_{43} & G_{44} & G_{45} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

$$G_{16} = y_2^1, \quad G_{26} = y_3^1, \quad G_{36} = y_4^1 - 1, \quad G_{46} = y_2^m.$$

Текст подпрограммы JACOB заполняется в соответствии с формулами для компонент вектора f :

$$\begin{aligned} f_1 &= q_2, \\ f_2 &= s_0 [(s_1 + s_2)q_1 - s_2q_3 - Q\varphi(q_1)q_4], \\ f_3 &= s_2(q_1 - q_2), \\ f_4 &= -s_3\varphi(q_1)q_4. \end{aligned}$$

В подпрограмме PRINT содержится вариант выдачи результатов на печать с обращением к подпрограмме SOLV-Q при $Q\emptyset = 2\emptyset\emptyset$. Заметим, что в силу (38) справедливо соотношение

$$Q = s_3 \frac{s_1\sigma + y_3(1)}{1 - y_4(1)}, \quad \sigma = \int_0^1 y_1(x) dx. \quad (39)$$

В данном примере множественность решений удобно представить графиком зависимости Q от σ . Таблица значений Q и σ (рис. 25) заполня-

BPR-Q

TWO POINT BOUNDARY VALUE PROBLEM :

0 < X < 1

$$Y' = V$$

$$V' = S0 * [(S1 + S2) * Y - S2 * W - Q * F(Y + S4) * Z]$$

$$W' = S2 * (Y - W)$$

$$Z' = -S3 * F(Y + S4) * Z$$

$$X=0 : V=W=0, Z=1$$

$$X=1 : V=0$$

$$F(U) = 1 / [S5 + \exp(-U / (1 + S6 * U))]$$

S0=10, S1=8, S2=5, S3=5, S4=-1, S5=.5, S6=.05

C9= 3.30192E-02

C9= 5.10511E-08

C9= 5.10511E-08

INTEG= 0 Q9= 0 Q= 0

Y(0)= 0 Y(1)= 0 W(1)= 0

Z(1)= .966981 Q' = 1257.14 I9= 6 J9= 1

C9= .302395

C9= 2.40618E-03

C9= 1.34811E-06

C9= 2.47475E-07

INTEG= .221958 Q9= 218.719 Q= 218.718

Y(0)= .16884 Y(1)= .25 W(1)= .242728

Z(1)= .953859 Q' = 566.832 I9= 6 J9= 1

C9= 1.57639E-04

C9= 2.48383E-07

C9= 1.38080E-07

INTEG= .195063 Q9= 199.976 Q= 199.976

Y(0)= .149637 Y(1)= .218674 W(1)= .212544

Z(1)= .955668 Q' = 630.664 I9= 6 J9= 1

C9= 1.97585E-07

C9= 3.33272E-08

INTEG= .195094 Q9= 199.999 Q= 199.999

Y(0)= .14966 Y(1)= .21871 W(1)= .212579

Z(1)= .955666 Q' = 630.587 I9= 6 J9= 1

X	Y(X)	V(X)	W(X)	Z(X)
0.0	0.14966	0.00000	0.00000	1.00000
0.2	0.17174	0.13334	0.10125	0.99146
0.4	0.19467	0.09120	0.15466	0.98267
0.6	0.20872	0.05153	0.18553	0.97370
0.8	0.21615	0.02433	0.20320	0.96468
1.0	0.21871	0.00000	0.21258	0.95567

DN E

Рис. 25.

ется в процессе продолжения решения по параметру. Интеграл σ вычисляется в подпрограмме PRINT по квадратурной формуле точности 4:

$$\sigma = \sum_{k=1}^{m-1} \frac{h_k}{2} \left\{ y_1(x_k) + y_1(x_{k+1}) + \frac{h_k}{6} [y_2(x_k) - y_2(x_{k+1})] \right\}.$$

При печати результатов используются обозначения формулировки краевой задачи (38), выдаваемой на печать в подпрограмме INPUT.

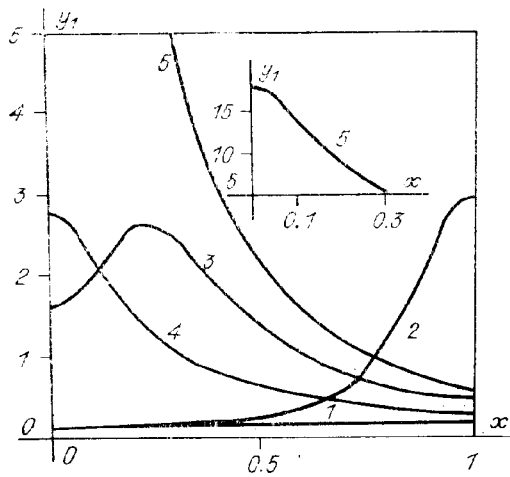


Рис. 26.

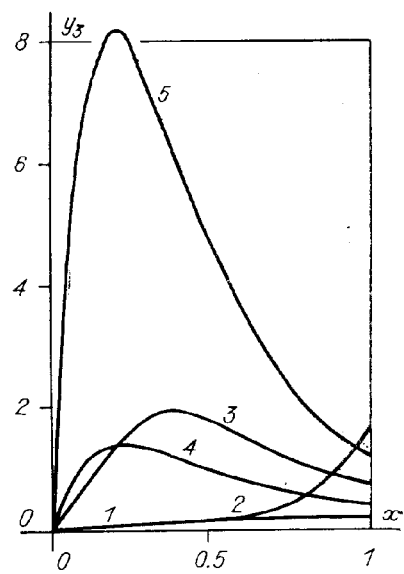


Рис. 27.

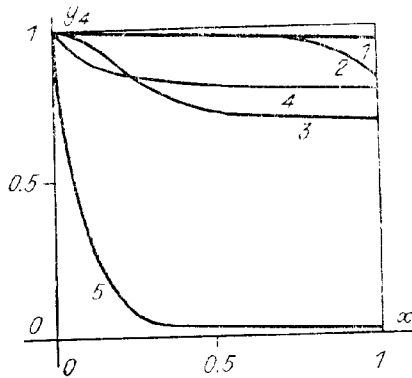


Рис. 28.

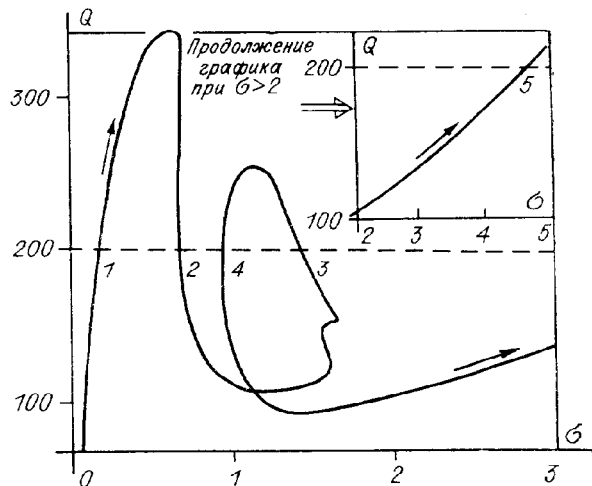


Рис. 29.

Кроме того, INTEG — обозначение σ , Q9 — вычисляемое значение параметра Q , а Q — значение, определяемое по формуле (39), C9 — норма невязки на итерациях (см. рис. 19). Сопоставление Q9 и Q служит косвенным контролем вычислений. Печать результатов завершается таблицей функций $y_1(x)$, $y_2(x)$, $y_3(x)$ и $y_4(x)$, являющихся решением (38) при $Q = 200$.

Дальнейшее продолжение решения по параметру показывает, что при $Q = 200$ краевая задача имеет пять решений. На рис. 26—28 представлены графики функций $y_1(x)$, $y_3(x)$ и $y_4(x)$ этих пяти решений. Общая картина множественности решений дана на рис. 29 в виде графика функций $Q(\sigma)$. Цифрами 1—5 отмечена последовательность решений (38) при $Q = 200$. В дальнейшем после σ , отвечающему пятому решению, график функции $Q(\sigma)$ характеризуется монотонным ростом.

Пример 3. В связи с описанием работы пленочного электростатического реле рассмотрим следующую краевую задачу:

$$s_0 \frac{d^4 u}{dx^4} = \frac{d^2 u}{dx^2} + \frac{Q}{(1-u)^2}, \quad 0 < x < 1,$$

$$\frac{du}{dx}(0) = \frac{d^3 u}{dx^3}(0) = u(1) = \frac{du}{dx}(1) = 0,$$

или

$$\frac{dy_1}{dx} = y_2, \quad \frac{dy_2}{dx} = y_3, \quad \frac{dy_3}{dx} = y_4, \quad \frac{dy_4}{dx} = \frac{1}{s_0} \left[y_3 + \frac{Q}{(1-y_1)^2} \right], \quad 0 < x < 1,$$

$$y_2 = y_4 = 0 \text{ при } x = 0, \quad y_1 = y_2 = 0 \text{ при } x = 1,$$

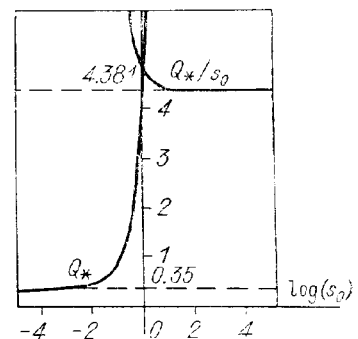


Рис. 30.

TWO POINT BOUNDARY VALUE PROBLEM :

$$\begin{aligned}
 0 < X < 1, \quad Y' &= U \\
 U' &= V \\
 V' &= W \\
 W' &= [V + Q/(1-Y) + 2]/S_0 \\
 X=0: \quad U=W=0; \quad X=1: \quad Y=U=0 \\
 S_0 &= .0001
 \end{aligned}$$

Y(0)=0.00000	Q9=0.00000	Q' = +2.04082E+00
S=0.00000E+00	I9=1.0	J9=1.0
Y(0)=0.10000	Q9=0.17139	Q' = +1.40037E+00
S=1.71389E+03	I9=1.0	J9=1.0
Y(0)=0.20000	Q9=0.28268	Q' = +8.39052E-01
S=2.82678E+03	I9=1.0	J9=1.0
Y(0)=0.30000	Q9=0.34184	Q' = +3.57716E-01
S=3.41840E+03	I9=1.0	J9=1.0
Y(0)=0.30000	Q9=0.34184	Q' = +3.57740E-01
S=3.41842E+03	I9=1.0	J9=1.0
Y(0)=0.40000	Q9=0.35693	Q' = -4.22863E-02
S=3.56934E+03	I9=1.0	J9=1.0
Y(0)=0.38838	Q9=0.35718	Q' = -4.93334E-05
S=3.57182E+03	I9=1.0	J9=1.0

X	Y(X)	U(X)	V(X)	W(X)
0.00000	0.38838	-0.00000	-9.54556E-01	+0.00000E+00
0.17746	0.37347	-0.16669	-9.09720E-01	+4.86154E-01
0.35492	0.33010	-0.31874	-7.95840E-01	+7.58714E-01
0.52237	0.26618	-0.44100	-6.63319E-01	+7.97216E-01
0.63238	0.21383	-0.50923	-5.77946E-01	+7.49610E-01
0.71655	0.16899	-0.55529	-5.17237E-01	+6.96738E-01
0.78325	0.13084	-0.58829	-4.72554E-01	+6.73441E-01
0.83703	0.09853	-0.61279	-4.36777E-01	+8.80418E-01
0.88649	0.06770	-0.63357	-3.83449E-01	+3.31520E+00
0.92665	0.04195	-0.64634	-3.22564E-02	+3.62228E+01
0.96333	0.01831	-0.61468	+4.54580E+00	+4.92133E+02
1.00000	-0.00000	+0.00000	+6.73588E+01	+6.77184E+03

DONE

Рис. 31.

где s_0 — параметр, пропорциональный жесткости гибкого электрода и обратно пропорциональный натяжению, Q — параметр, пропорциональный квадрату разности потенциалов между электродами. Очевидно, при $s_0 \rightarrow 0$ мы получаем краевую задачу (5). Определим значения Q , при которых имеет место ветвление решений (40) в зависимости от s_0 .

В данном случае $n=4$, $n_0=2$. В соответствии с формулировкой краевой задачи (40)

$$\begin{bmatrix} G_{11} & G_{12} & G_{13} & G_{14} & G_{15} & G_{16} \\ G_{21} & G_{22} & G_{23} & G_{24} & G_{25} & G_{26} \\ G_{31} & G_{32} & G_{33} & G_{34} & G_{35} & G_{36} \\ G_{41} & G_{42} & G_{43} & G_{44} & G_{45} & G_{46} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & y_2^1 \\ 0 & 0 & 0 & 1 & 0 & y_4^1 \\ 1 & 0 & 0 & 0 & 0 & y_1^m \\ 0 & 1 & 0 & 0 & 0 & y_2^m \end{bmatrix}$$

Кроме того, пользователем заполняются подпрограммы применительно к правым частям (40): $f_1 = q_2$, $f_2 = q_3$, $f_3 = q_4$, $f_4 = \frac{1}{s_0} \left[q_3 + \frac{Q}{(1 - q_1)^2} \right]$

(см. приложение к § 8).

Как и в (5), график функции $\lambda = y_1(0, Q, s_0)$, определенный на решениях краевой задачи (40), имеет качественно тот же вид, что и на рис. 2. Иначе говоря, для каждого s_0 имеется максимальное значение $Q = Q_*$, при котором решение (40) еще существует. Зависимость Q_* от s_0 дана на рис. 30, из которой следует, что при $s_0 \rightarrow 0$ ($\log(s_0) \rightarrow -\infty$) значение Q_* стремится к 0.35.

Для наглядного представления другого предельного случая, когда $s_0 \rightarrow \infty$, на рис. 30 приведена зависимость Q_*/s_0 от s_0 . При $s_0 > 1$ это отношение быстро стремится к 4.381, что является максимальным значением Q_* краевой задачи

$$\frac{dy_1}{dx} = y_2, \frac{dy_2}{dx} = y_3, \frac{dy_3}{dx} = y_4, \frac{dy_4}{dx} = \frac{Q}{(1 - y_1)^2}, \quad 0 < x < 1, \quad (41)$$

$$y_2 = y_4 = 0 \text{ при } x = 0, \quad y_1 = y_2 = 0 \text{ при } x = 1.$$

Вычисление Q_* в краевых задачах (40) и (41) проводилось с использованием подпрограммы EXTREM. Таким образом, в данном примере удалось численно исследовать зависимость Q_* от s_0 при всех значениях $s_0 \geq 0$.

В качестве теста на рис. 31 приведен вариант выдачи на печать результатов решения (40) при $s_0 = 10^{-4}$. Значения входных параметров содержатся в подпрограмме INPUT. Предложение с номером 3060 подпрограммы PRINT — обращение к подпрограмме EXTREM. Работа программы завершается печатью таблицы значений искомых функций в узлах сформированной сетки по x при обращении к подпрограмме DELTA к моменту, когда $Q = Q_* = 0.3572$.

§ 9. ДВУХТОЧЕЧНАЯ КРАЕВАЯ ЗАДАЧА С КРАЕВЫМИ УСЛОВИЯМИ ОБЩЕГО ВИДА

В этом параграфе рассматривается двухточечная краевая задача для системы обыкновенных дифференциальных уравнений с более общими, чем в (1), краевыми условиями:

$$\frac{dy}{dx} = f(x, y, Q), \quad x \in (a, b), \quad (42)$$

$$g(y(a), y(b)) = 0.$$

Здесь g — достаточно гладкая вектор-функция своих аргументов, задающая n независимых краевых условий. В частности, к формулировке (42) сводится краевая задача, описывающая периодическое решение с заданным периодом.

Дискретная модель (42) при использовании обозначений § 2 имеет вид (см. (10)):

$$g(y^1, y^m, Q) = 0, \\ F^i = y^i - y^{i+1} + \frac{h_i}{6} [f(x_i, y^i, Q) + 4f(\bar{x}_i, \bar{y}^i, Q) + \\ + f(x_{i+1}, y^{i+1}, Q)] = 0, \quad i = 1, 2, \dots, m - 1.$$

Мы по-прежнему будем использовать (2) в качестве векторной записи полученной системы трансцендентных уравнений, полагая в дальнейшем,

что

$$\Phi(Y, Q) = \begin{bmatrix} g(y^1, y^m, Q) \\ F^1(y^1, y^2, Q) \\ \dots \\ F^{(m-1)}(y^{m-1}, y^m, Q) \end{bmatrix}. \quad (43)$$

Очевидно, (43) отвечает матрица якобиана следующей структуры:

$$[\Phi_Y, \Phi_Q] = \begin{bmatrix} \Phi_Y & \Phi_Q \\ \begin{array}{ccc|c} g_{y^1} & & & g_{y^m} \\ F_{y^1}^1 & F_{y^2}^1 & & 0 \\ & F_{y^2}^2 & F_{y^3}^2 & 0 \\ \hline & & & \\ & F_{y^{m-2}}^{m-2} & F_{y^{m-1}}^{m-2} & 0 \\ & & F_{y^{m-1}}^{m-1} & F_{y^m}^{m-1} \end{array} & \begin{array}{c} g \\ F_Q^1 \\ F_Q^2 \\ \hline F_Q^{m-2} \\ F_Q^{m-1} \end{array} \end{bmatrix}. \quad (44)$$

Проблема состоит в использовании метода продолжения решения по параметру системы $\Phi[Y, Q] = 0$ с применением параметризации, рассмотренной в § 3. При этом учитывается вид матрицы (44). Заметим, что изменения в программе BPR-Q коснутся лишь программы SOLVE. Остальные особенности легко учитываются при заполнении «нестандартных» блоков, заполняемых пользователем. Это относится к подпрограммам INPUT и BOND.

Так как в принятой последовательности уравнений системы $\Phi(Y, Q) = 0$ первые n мест занимают уравнения, отвечающие краевым условиям (42), то в подпрограмме INPUT следует задать $n_0 = n$.

В подпрограмме BOND используется массив с элементами G_{ij} , размеры которого теперь равны $N \times (2N + 2)$. Элементам G_{ij} присваиваются следующие значения:

$$G_{i1} = \frac{\partial g_i}{\partial y_1^1}, G_{i2} = \frac{\partial g_i}{\partial y_2^1}, \dots, G_{in} = \frac{\partial g_i}{\partial y_n^1}, G_{i n+1} = \frac{\partial g_i}{\partial y_1^m}, G_{i n+2} = \frac{\partial g_i}{\partial y_2^m}, \dots$$

$$\dots, G_{i 2n} = \frac{\partial g_i}{\partial y_n^m}, G_{i 2n+1} = \frac{\partial g_i}{\partial Q}, G_{i 2n+2} = g_i, i = 1, 2, \dots, n.$$

Этим исчерпываются замечания относительно заполнения «нестандартных» блоков BPR-Q.

Остановимся на изменениях в программе SOLVE. Положив $n_0 = n$, мы должны учесть наличие блока g_{y^m} матрицы $[\Phi_Y, \Phi_Q]$ (см. (44)). Это относится как к процедуре формирования матрицы $\Phi_s(S, \mu)$, так и к реализации прямого и обратного ходов прогонки.

Измененная программа SOLVE под именем SOLV включена в общую программу в качестве блока, обращение к которому происходит при $M = N$ в программе MATRIC:

MATRIC

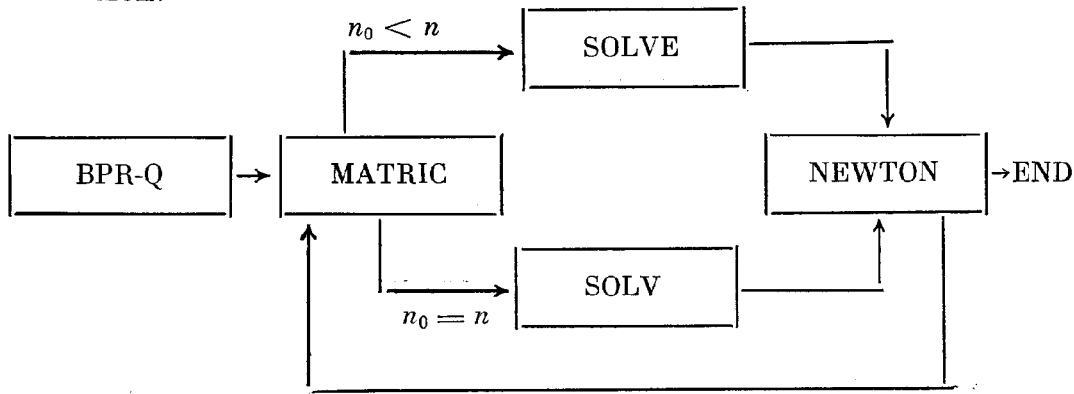
164 IF $M = N$ THEN 166

165 CHAIN SOLVE

166 CHAIN SOLV

Таким образом, предлагаемая программа характеризуется следующей

блок-схемой:



Включение в блок-схему программы SOLVE позволяет более экономично строить решение краевой задачи (1), чем при использовании общего подхода, когда (1) представляется в виде краевой задачи (42).

Принципиально организация прямого и обратного ходов прогонки в программе SOLV остается той же, что и в программе SOLVE. Остановимся на некоторых подробностях.

Описание SOLV удобно, как и в § 4, связать с примером. Пусть $n_0 = n = 3$, $m = 4$. Тогда элементы матрицы $\Omega = [\Phi_Y, \Phi_Q, \Phi]$ схематично можно представить в виде

$$\Omega = \begin{bmatrix} \begin{array}{cc} \Phi_Y & \Phi_Q, \Phi \end{array} \\ \begin{array}{ccc|cc} \hline GGG & & & GGG & G & G \\ GGG & & & GGG & G & G \\ GGG & & & GGG & G & G \\ \hline AAA & AAA & & & Q & F \\ AAA & AAA & & & Q & F \\ AAA & AAA & & & Q & F \\ \hline & AAA & AAA & & Q & F \\ & AAA & AAA & & Q & F \\ & AAA & AAA & & Q & F \\ \hline & & AAA & AAA & Q & F \\ & & AAA & AAA & Q & F \\ & & AAA & AAA & Q & F \\ \hline \end{array} \end{bmatrix}. \quad (45)$$

При организации прямого хода прогонки используется вспомогательный прямоугольный массив w размеров $L_1 \times L_2$, причем максимальные значения L_1 и L_2 равны $2n + 1$ и $2n + 3$ соответственно. Кроме того, для запоминания преобразуемых столбцов матрицы Ω , содержащих в стартовой позиции матрицу g_{ym} , используется прямоугольный массив C размеров $(m - 1) \times n^2$. Учет структуры матрицы Ω , определяемой заданием индексов i_* и j_* текущего параметра μ , проявляется при преобразовании Ω по формулам метода Гаусса (подпрограмма GAUSS), т. е. в отслеживании зависимости L_1 , L_2 и L_3 от номера i шага прямого хода прогонки, $i = 1, 2, \dots, m - 1$. Обозначим через $\Omega^{[i]}$ результат преобразования исходной матрицы $\Omega^{[0]}$ на i -м шаге. Характерной для описания процесса является матрица $\Omega^{[m-2]}$ на предпоследнем шаге. Обратимся к (45).

(а) При $i_* = m + 1$, $j_* = 0$ матрица $\Omega^{[0]} = \Omega$ преобразуется в матрицу $\Omega^{[2]}$ следующего вида:

$$\Omega^{[2]} = \begin{bmatrix} \begin{array}{ccc|cc} \hline AAA & AAA & & CCC & Q & F \\ AA & AAA & & CCC & Q & F \\ A & AAA & & CCC & Q & F \\ \hline & AAA & AAA & CCC & Q & F \\ & AA & AAA & CCC & Q & F \\ & A & AAA & CCC & Q & F \\ \hline \end{array} \\ \begin{array}{ccc|cc} \hline & XXX & XXX & X & X \\ & XXX & XXX & X & X \\ & XXX & XXX & X & X \\ \hline & AAA & AAA & Q & F \\ & AAA & AAA & Q & F \\ & AAA & AAA & Q & F \\ \hline \end{array} \end{bmatrix}. \quad (46)$$

Завершает прямой ход прогонки преобразование массива w матрицы $\Omega^{[2]}$ (46):

$$w \Rightarrow \tilde{w} = \begin{bmatrix} \text{XXX} & \text{XXX} & \text{XX} \\ & \text{XX} & \text{XXX} & \text{XX} \\ & & \text{X} & \text{XXX} & \text{XX} \\ & & & \text{XXX} & \text{XX} \\ & & & & \text{XX} & \text{XX} \\ & & & & & \text{X} & \text{XX} \end{bmatrix},$$

после чего матрица $\tilde{\Phi}$ становится верхней треугольной.

В общем случае $\Omega^{[m-1]}$ имеет вид

$$\Omega^{[m-1]} = \begin{bmatrix} A^{[1]} & B^{[1]} & & & C^{[1]} & Q^{[1]} & F^{[1]} \\ & A^{[2]} & B^{[2]} & & C^{[2]} & Q^{[2]} & F^{[2]} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ & & & A^{[m-2]} & B^{[m-2]} & C^{[m-2]} & Q^{[m-2]} & F^{[m-2]} \\ & & & & \boxed{\text{массив } \tilde{w}} & & & \end{bmatrix}. \quad (47)$$

Здесь $A^{[i]}$, $B^{[i]}$ — элементы блочных диагоналей (см. рис. (25)), $C^{[i]}$ — элементы блочного столбца, $Q^{[i]}$, $F^{[i]}$ — векторы, составляющие $\tilde{\Phi}_Q$ и Φ соответственно, i — номер блочной строки, \tilde{w} — преобразованный по формулам Гаусса массив w на последнем шаге прогонки. Напомним, что L_1 , L_2 и L_3 — параметры, характеризующие трапециевидную структуру матрицы \tilde{w} , к которой преобразуются w на каждом шагу прямого хода прогонки. Структура матрицы $\Omega^{[2]}$ (см. (46)) подсказывает, что

$$L_1 = 2n, \quad L_2 = 3n + 2, \quad L_3 = n, \quad 1 \leq i \leq m - 2,$$

$$L_1 = 2n, \quad L_2 = 2n + 2, \quad L_3 = 2n, \quad i = m - 1.$$

Пусть, как и ранее, размеры $A^{[i]}$ и $B^{[i]}$ суть $L_4 \times L_4$ и $L_4 \times L_5$ соответственно. Кроме того, размеры $C^{[i]}$ — $L_4 \times L_6$. Имеем $L_4 = L_5 = L_6 = n$, $1 \leq i \leq m - 2$. Обозначим через Z составной вектор, состоящий из векторов Z^1, Z^2, \dots, Z^m , который является решением системы $\tilde{\Phi}_Y Z = -\Phi$. Согласно (47) на первом шаге обратного хода прогонки определяются векторы Z^m и Z^{m-1} . Остальные Z^i , $i = m - 2, m - 3, \dots, 1$, находятся последовательно по рекуррентным формулам $A^{[i]} Z^i = -F^{[i]} - B^{[i]} Z^{i+1} - C^{[i]} Z^m$.

(б) При $i_* = 1$, $1 \leq j_* \leq n$

$$\Omega^{[0]} = \begin{bmatrix} & & & & \Phi_S & & & & & & \Phi & \Phi_\mu \\ \text{GG} & \text{O} & & & & & \text{GGG} & \text{G} & & & \text{G} & \text{G} \\ \text{GG} & \text{O} & & & & & \text{GGG} & \text{G} & & & \text{G} & \text{G} \\ \text{GG} & \text{O} & & & & & \text{GGG} & \text{G} & & & \text{G} & \text{G} \\ \text{AA} & \text{AAA} & \text{O} & & & & & \text{Q} & & & \text{F} & \text{A} \\ \text{AA} & \text{AAA} & \text{O} & & & & & \text{Q} & & & \text{F} & \text{A} \\ \text{AA} & \text{AAA} & \text{O} & & & & & \text{Q} & & & \text{F} & \text{A} \\ & & & \text{AAA} & \text{AAA} & & & \text{Q} & & & \text{F} & \text{O} \\ & & & \text{AAA} & \text{AAA} & & & \text{Q} & & & \text{F} & \text{O} \\ & & & \text{AAA} & \text{AAA} & & & \text{Q} & & & \text{F} & \text{O} \\ & & & & & \text{AAA} & \text{AAA} & \text{Q} & & & \text{F} & \text{O} \\ & & & & & \text{AAA} & \text{AAA} & \text{Q} & & & \text{F} & \text{O} \\ & & & & & \text{AAA} & \text{AAA} & \text{Q} & & & \text{F} & \text{O} \end{bmatrix}.$$

Как легко заметить, имеет место следующая зависимость L_1, L_2 и L_3 от i :

$$L_1 = 2n, \quad L_2 = 2n + 3, \quad L_3 = n, \quad 1 \leq i \leq m - 2,$$

$$L_1 = 2n, \quad L_2 = 2n + 2, \quad L_3 = 2n, \quad i = m - 1.$$

Кроме того, $L_4 = n, L_5 = n, L_6 = n - 1, 1 \leq i \leq m - 2$. Обратный ход прогонки организуется точно так же, как и в предыдущем случае.

(d) Пусть $1 < i_* < m - 1, 1 \leq j_* \leq n$. Для конкретности положим $i_* = 2$. При этом матрица $\Omega^{[0]}$ имеет вид

$$\Omega^{[0]} = \left[\begin{array}{c|c|c|c|c} & \Phi_S & & \Phi & \Phi_\mu \\ \hline & \begin{array}{cc} \text{GGG} & \text{GGG} \\ \text{GGG} & \text{GGG} \\ \text{GGG} & \text{GGG} \end{array} & \begin{array}{c} \text{G} \\ \text{G} \\ \text{G} \end{array} & \begin{array}{c} \text{G} \\ \text{G} \\ \text{G} \end{array} & \begin{array}{c} \text{O} \\ \text{O} \\ \text{O} \end{array} \\ \hline & \begin{array}{cc} \text{AAA} & \text{AA} \\ \text{AAA} & \text{AA} \\ \text{AAA} & \text{AA} \end{array} & \begin{array}{c} \text{Q} \\ \text{Q} \\ \text{Q} \end{array} & \begin{array}{c} \text{F} \\ \text{F} \\ \text{F} \end{array} & \begin{array}{c} \text{A} \\ \text{A} \\ \text{A} \end{array} \\ \hline & & \begin{array}{cc} \text{AA} & \text{AAA} \\ \text{AA} & \text{AAA} \\ \text{AA} & \text{AAA} \end{array} & \begin{array}{c} \text{Q} \\ \text{Q} \\ \text{Q} \end{array} & \begin{array}{c} \text{F} \\ \text{F} \\ \text{F} \end{array} & \begin{array}{c} \text{A} \\ \text{A} \\ \text{A} \end{array} \\ \hline & & & \begin{array}{cc} \text{AAA} & \text{AAA} \\ \text{AAA} & \text{AAA} \\ \text{AAA} & \text{AAA} \end{array} & \begin{array}{c} \text{Q} \\ \text{Q} \\ \text{Q} \end{array} & \begin{array}{c} \text{F} \\ \text{F} \\ \text{F} \end{array} & \begin{array}{c} \text{O} \\ \text{O} \\ \text{O} \end{array} \end{array} \right].$$

Приведем соответствующую $\Omega^{[0]}$ матрицу $\Omega^{[2]}$: $\Omega^{[0]} \Rightarrow \Omega^{[2]}$,

$$\Omega^{[2]} = \left[\begin{array}{c|c|c|c|c|c} \text{AAA} & \text{AA} & & \text{CCC} & \text{Q} & \text{F} & \text{R} \\ \text{AA} & \text{AA} & & \text{CCC} & \text{Q} & \text{F} & \text{R} \\ \text{A} & \text{AA} & & \text{CCC} & \text{Q} & \text{F} & \text{R} \\ & \text{AA} & \text{AAA} & \text{CCC} & \text{Q} & \text{F} & \text{R} \\ & \text{A} & \text{AAA} & \text{CCC} & \text{Q} & \text{F} & \text{R} \\ \hline & & \text{XXX} & \text{XXX} & \text{X} & \text{X} & \text{X} \\ & & \text{XXX} & \text{XXX} & \text{X} & \text{X} & \text{X} \\ & & \text{XXX} & \text{XXX} & \text{X} & \text{X} & \text{X} \\ & & \text{XXX} & \text{XXX} & \text{X} & \text{X} & \text{X} \\ \hline \text{W} & & \text{AAA} & \text{AAA} & \text{Q} & \text{F} & \text{O} \\ & & \text{AAA} & \text{AAA} & \text{Q} & \text{F} & \text{O} \\ & & \text{AAA} & \text{AAA} & \text{Q} & \text{F} & \text{O} \end{array} \right].$$

Отметим, что в данном случае

$$L_1 = 2n, \quad L_2 = 3n + 3, \quad L_3 = n, \quad 1 \leq i < i_*,$$

$$L_1 = 2n, \quad L_2 = 3n + 3, \quad L_3 = n - 1, \quad i = i_*,$$

$$L_1 = 2n + 1, \quad L_2 = 3n + 3, \quad L_3 = n, \quad i_* < i \leq m - 2,$$

$$L_1 = 2n + 1, \quad L_2 = 2n + 3, \quad L_3 = 2n + 1, \quad i = m - 1,$$

$$L_4 = n, \quad L_5 = n, \quad L_6 = n, \quad 1 \leq i \leq i_* - 1,$$

$$L_4 = n - 1, \quad L_5 = n, \quad L_6 = n, \quad i = i_*,$$

$$L_4 = n, \quad L_5 = n, \quad L_6 = n, \quad i_* < i \leq m - 2.$$

Обратный ход прогонки организуется точно так же, как и в (b).

(e) Наконец, при $i_* = m - 1, 1 \leq j_* \leq n$ имеем

$$\Omega^{[0]} = \left[\begin{array}{c|c|c|c|c} & \Phi_S & & \Phi & \Phi_\mu \\ \hline & \begin{array}{cc} \text{GGG} & \text{GGG} \\ \text{GGG} & \text{GGG} \\ \text{GGG} & \text{GGG} \end{array} & \begin{array}{c} \text{Q} \\ \text{Q} \\ \text{Q} \end{array} & \begin{array}{c} \text{G} \\ \text{G} \\ \text{G} \end{array} & \begin{array}{c} \text{O} \\ \text{O} \\ \text{O} \end{array} \\ \hline & \begin{array}{cc} \text{AAA} & \text{AAA} \\ \text{AAA} & \text{AAA} \\ \text{AAA} & \text{AAA} \end{array} & \begin{array}{c} \text{Q} \\ \text{Q} \\ \text{Q} \end{array} & \begin{array}{c} \text{F} \\ \text{F} \\ \text{F} \end{array} & \begin{array}{c} \text{O} \\ \text{O} \\ \text{O} \end{array} \\ \hline & & \begin{array}{cc} \text{AAA} & \text{AA} \\ \text{AAA} & \text{AA} \\ \text{AAA} & \text{AA} \end{array} & \begin{array}{c} \text{Q} \\ \text{Q} \\ \text{Q} \end{array} & \begin{array}{c} \text{F} \\ \text{F} \\ \text{F} \end{array} & \begin{array}{c} \text{A} \\ \text{A} \\ \text{A} \end{array} \\ \hline & & & \begin{array}{cc} \text{AA} & \text{AAA} \\ \text{AA} & \text{AAA} \\ \text{AA} & \text{AAA} \end{array} & \begin{array}{c} \text{Q} \\ \text{Q} \\ \text{Q} \end{array} & \begin{array}{c} \text{F} \\ \text{F} \\ \text{F} \end{array} & \begin{array}{c} \text{A} \\ \text{A} \\ \text{A} \end{array} \end{array} \right].$$

$$\Omega^{[0]} \Rightarrow \Omega^{[2]},$$

$$\Omega^{[2]} = \left[\begin{array}{cccccc} \text{AAA} & \text{AAA} & & & \text{Q} & \text{F} & \text{R} \\ & \text{AA} & \text{AAA} & & \text{Q} & \text{F} & \text{R} \\ & & \text{A} & \text{AAA} & \text{Q} & \text{F} & \text{R} \\ & & & \text{AAA} & \text{AA} & \text{Q} & \text{F} & \text{R} \\ & & & \text{AA} & \text{AA} & \text{Q} & \text{F} & \text{R} \\ & & & \text{A} & \text{AA} & \text{Q} & \text{F} & \text{R} \\ \hline & & & \text{XX} & \text{XXX} & \text{X} & \text{X} & \text{X} \\ & & & \text{XX} & \text{XXX} & \text{X} & \text{X} & \text{X} \\ & & & \text{XX} & \text{XXX} & \text{X} & \text{X} & \text{X} \\ & & & \text{AA} & \text{AAA} & \text{Q} & \text{F} & \text{A} \\ & & & \text{AA} & \text{AAA} & \text{Q} & \text{F} & \text{A} \\ & & & \text{AA} & \text{AAA} & \text{Q} & \text{F} & \text{A} \end{array} \right].$$

Интересующие нас размеры массивов в зависимости от номера шага i прямого хода прогонки следующие:

$$L_1 = 2n, \quad L_2 = 3n + 3, \quad L_3 = n, \quad 1 \leq i \leq m - 2,$$

$$L_1 = 2n, \quad L_2 = 2n + 2, \quad L_3 = 2n, \quad i = m - 1,$$

$$L_4 = n, \quad L_5 = n, \quad L_6 = n, \quad 1 \leq i \leq m - 2.$$

Обратный ход прогонки организуется точно так же, как и в (b).

Текст программы SOLV дан в приложении к § 9. Обратим внимание на изменение размеров массива $G[I, J]$, заполняемого в подпрограмме BOND программы MATRIC — $G[N\emptyset, 2N\emptyset + 2]$. Это следует учесть в предложенных COM программ MATRIC, SOLV и NEWTON. Кроме того, в программе SOLV используется массив $B[I, J]$ (массив C), размеры массива — $B[N - 1, N\emptyset^2]$. Размеры массива $W[I, J]$ — $W[2N\emptyset + 1, 3N\emptyset + 3]$.

Приведем краткие пояснения к программе SOLV (см. также обозначения к программе SOLVE):

1 $\emptyset\emptyset\emptyset$ —151 \emptyset — формирование матрицы $\Omega^{[0]}$

152 \emptyset —2 \emptyset 1 \emptyset — прямой ход прогонки, формирование матрицы $\Omega^{[m-2]}$

2 \emptyset 2 \emptyset —23 $\emptyset\emptyset$ — формирование массива w , соответствующего $\Omega^{[m-2]}$

231 \emptyset —3 \emptyset 3 \emptyset — обратный ход прогонки

3 \emptyset 5 \emptyset —326 \emptyset — подпрограмма GAUSS.

§ 10. ЧИСЛЕННЫЕ ПРИМЕРЫ

Пример 1. Как уже отмечалось, численное решение краевой задачи (1) может быть найдено с использованием программы SOLV. Для этого достаточно краевые условия (1) записать в формулировке (42), положив $n_0 = n$. В связи с этим рассмотрим (38), где $n = 4$. Задав $n_0 = 4$, сохраним значения остальных параметров подпрограммы INPUT (см. приложение к § 8). Тем самым выполнено условие обращения к подпрограмме SOLV: $M\emptyset = N\emptyset = 4$. Заметим, что размеры массивов программы SOLV, текст которой приведен в приложении к § 9, отвечают рассматриваемому примеру.

В обозначениях § 9 краевые условия (38) имеют вид

$$g_1 = y_2^1, \quad g_2 = y_3^1, \quad g_3 = y_4^1 - 1, \quad g_4 = y_2^m.$$

Отсюда следует задание элементов массива G_{ij} подпрограммы BOND:

$$\left[\begin{array}{cccccccccc} G_{11} & G_{12} & G_{13} & G_{14} & G_{15} & G_{16} & G_{17} & G_{18} & G_{19} & G_{110} \\ G_{21} & G_{22} & G_{23} & G_{24} & G_{25} & G_{26} & G_{27} & G_{28} & G_{29} & G_{210} \\ G_{31} & G_{32} & G_{33} & G_{34} & G_{35} & G_{36} & G_{37} & G_{38} & G_{39} & G_{310} \\ G_{41} & G_{42} & G_{43} & G_{44} & G_{45} & G_{46} & G_{47} & G_{48} & G_{49} & G_{410} \end{array} \right] =$$

$$= \left[\begin{array}{cccccccccc} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & G_{110} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & G_{210} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & G_{310} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & G_{410} \end{array} \right],$$

$$G_{110} = y_2^1, \quad G_{210} = y_3^1, \quad G_{310} = y_4^1 - 1, \quad G_{410} = y_2^m.$$

Тексты подпрограмм INPUT и BOND в связи с (38) приведены в приложении к § 9. Тексты остальных подпрограмм, заполняемых пользователем, остаются теми же, что и в приложении к § 8, в том числе и текст подпрограммы PRINT. При этом выдача результатов на печать полностью повторяется.

Пример 2. Рассмотрим краевую задачу, описывающую нелинейные колебания с заданным периодом T :

$$\frac{d^2y}{dt^2} + s_0^2 y = Qy^2 + s_9 \sin(s_1 t), \quad 0 \leq t \leq T, \quad T = 2\pi/s_1, \quad (48)$$

$$y(0) = y(T), \quad \frac{dy}{dt}(0) = \frac{dy}{dt}(T),$$

где s_0, s_1, s_9, Q — положительные параметры.

В качестве теста найдем численное решение (48) при $s_0 = \sqrt{2}$, $s_1 = s_9 = 1$, $Q = 0.2$ и сопоставим его с решением в виде степенного ряда по Q , в котором учтены три слагаемых:

$$y(t) = u_0(t) + Qu_1(t) + Q^2u_2(t) + \dots \quad (49)$$

где

$$u_0(t) = a_0 \sin(s_1 t), \quad u_1(t) = \frac{a_0^2}{2} [1/s_0^2 + a_1 \cos(2s_1 t)], \quad u_2(t) = \\ = \frac{a_0^3}{2} [a_2 \sin(s_1 t) + a_3 \sin(3s_1 t)],$$

$$a_0 = \frac{s_9}{s_0^2 - s_1^2}, \quad a_1 = \frac{1}{4s_1^2 - s_0^2}, \quad a_2 = \frac{2 - s_0^2 a_1}{s_0^2 (s_0^2 - s_1^2)}, \quad a_3 = \frac{a_1}{s_0^2 - 9s_1^2}.$$

Представим в (48) в стандартном виде (42):

$$t = Tx, \quad 0 < x < 1,$$

$$\frac{dy_1}{dx} = f_1 = Ty_2, \quad (50)$$

$$\frac{dy_2}{dx} = f_2 = T [Qy_1^2 + s_9 \sin(2\pi x) - s_0^2 y_1],$$

$$g_1 = y_1(0) - y_1(1) = 0, \quad g_2 = y_2(0) - y_2(1) = 0.$$

Тогда элементы массива G_{ij} (подпрограмма BOND) вычисляются по формулам

$$\begin{bmatrix} G_{11} & G_{12} & G_{13} & G_{14} & G_{15} & G_{16} \\ G_{21} & G_{22} & G_{23} & G_{24} & G_{25} & G_{26} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & G_{16} \\ 0 & 1 & 0 & -1 & 0 & G_{26} \end{bmatrix}, \\ G_{16} = y_1^1 - y_1^m, \quad G_{26} = y_2^1 - y_2^m.$$

Выражения элементов остальных массивов (подпрограммы JASOB и ADAPT) выписываются в полном соответствии с § 6, п. 2:

$$\xi = x_k, \quad q_i = y_i^k, \quad f_i = f_i(\xi, q_1, q_2, \dots, q_n), \quad i = 1, 2, \\ O_1 = f_1 = Tq_2, \quad R_1 = \frac{\partial f_1}{\partial Q} = 0, \quad u_{11} = \frac{\partial f_1}{\partial q_1} = 0, \quad u_{12} = \\ = \frac{\partial f_1}{\partial q_2} = T, \quad u_{13} = \frac{\partial f_1}{\partial \xi} = 0,$$

$$O_2 = f_2 = T [Qq_1^2 + s_9 \sin(2\pi\xi) - s_0^2 q_1], \quad u_{21} = \frac{\partial f_2}{\partial q_1} = T [2Qq_1 - s_0^2],$$

$$u_{22} = \frac{\partial f_2}{\partial q_2} = 0, \quad u_{23} = \frac{\partial f_2}{\partial \xi} = 2\pi T s_9 \cos(2\pi\xi), \quad R_2 = \frac{\partial f_2}{\partial Q} = Tq_1^2.$$

BPR-Q

TWO POINT BOUNDARY VALUE PROBLEM :

$$0 < X < 1$$

$$Y' = T * V, \quad P0 = 6.28319, \quad T = P0 / S1$$

$$V' = T * [S9 * SIN(P0 * X) - S0 + 2 * Y + Q * Y + 2]$$

$$Y(0) = Y(1), \quad V(0) = V(1)$$

$$S0 = SQR(2), \quad S1 = 1, \quad S9 = 1$$

$$C9 = 5.00025$$

$$C9 = .107487$$

$$C9 = 1.23518E-03$$

$$C9 = 2.38731E-07$$

$$C9 = 1.88096E-07$$

$$Q = .2$$

$$AMPL = 1.06274$$

$$X0 = .75$$

$$V0 = 5.14558E-02 \quad I9 = 22 \quad J9 = 0 \quad Q' = 1$$

X	Y(X)	U(X)
0.00	+0.10208	+0.10000
0.05	+0.40354	+0.40140
0.10	+0.65977	+0.65776
0.15	+0.85296	+0.85121
0.20	+0.97244	+0.97096
0.25	+1.01280	+1.01143
0.30	+0.97244	+0.97096
0.35	+0.85296	+0.85121
0.40	+0.65976	+0.65775
0.45	+0.40354	+0.40140
0.50	+0.10207	+0.10000
0.55	-0.21868	-0.22050
0.60	-0.52548	-0.52686
0.65	-0.78130	-0.78212
0.70	-0.95152	-0.95186
0.75	-1.01129	-1.01143
0.80	-0.95152	-0.95186
0.85	-0.78130	-0.78211
0.90	-0.52548	-0.52685
0.95	-0.21868	-0.22050
1.00	+0.10208	+0.10000

DONE

Рис. 32.

Значения входных параметров задачи приведены в подпрограмме INPUT. Отметим, что здесь Q играет роль текущего параметра системы в стартовой позиции, $Q = 0.2$, а в качестве начального приближения взяты нулевые сеточные значения $y_1(x)$ и $y_2(x)$ (см. (50)). Тексты нестандартных подпрограмм приведены в приложении к § 10.

Печать результатов (рис. 32) содержит значения нормы невязок $C9$ на каждой из итераций. В подпрограмме PRINT предусмотрена выдача значений Q , амплитуды колебаний ρ , $\rho = \max_x |y_1(x) - y_c|$ (обозначение $AMPL$), $X0$ — аргумент ρ , y_c (обозначение $V0$) — среднее значение $y_1(x)$, $y_c = \frac{1}{T} \int_0^T y_1(x) dx$, индексы $I9$, $J9$ заданного текущего параметра, Q' — производная Q по текущему параметру, а также таблица сеточных значений x , $y_1(x)$ (обозначение $Y(x)$) и приближенных значений $y_1(x)$

BPR-Q

TWO POINT BOUNDARY VALUE PROBLEM :

$$0 < X < 1$$

$$Y' = T * V, \quad P0 = 6.28319, \quad T = P0 / S1$$

$$V' = T * [S9 * SIN(P0 * X) - S0 + 2 * Y + Q * Y + 2]$$

$$Y(0) = Y(1), \quad V(0) = V(1)$$

$$S0 = SQR(2), \quad S1 = 1, \quad S9 = 1$$

$$Q = 0 \quad AMPL = 1.00004 \quad X0 = .75 \\ V0 = 4.09782E-08 \quad I9 = 21 \quad J9 = 1 \quad Q' = 2.00049$$

$$Q = .369042 \quad AMPL = 1.14448 \quad X0 = .75 \\ V0 = .102915 \quad I9 = 10 \quad J9 = 1 \quad Q' = 1.53367$$

$$Q = .568075 \quad AMPL = 1.3448 \quad X0 = .75 \\ V0 = .2053 \quad I9 = 3 \quad J9 = 1 \quad Q' = .591926$$

$$Q = .626101 \quad AMPL = 1.54976 \quad X0 = .75 \\ V0 = .297457 \quad I9 = 6 \quad J9 = 1 \quad Q' = .110166$$

$$Q = .628645 \quad AMPL = 1.74091 \quad X0 = .75 \\ V0 = .37896 \quad I9 = 6 \quad J9 = 1 \quad Q' = -.055093$$

$$Q = .628683 \quad AMPL = 1.74095 \quad X0 = .750012 \\ V0 = .378994 \quad I9 = 5 \quad J9 = 1 \quad Q' = -5.51246E-02$$

$$Q = .631159 \quad AMPL = 1.65457 \quad X0 = .750012 \\ V0 = .34251 \quad I9 = 5 \quad J9 = 1 \quad Q' = 4.23635E-04$$

$$Q = .631159 \quad AMPL = 1.65512 \quad X0 = .750012 \\ V0 = .342741 \quad I9 = 5 \quad J9 = 1 \quad Q' = -6.66466E-06$$

X	Y(X)	Y'(X)
0.00000	+0.59309	+1.28026
0.04698	+0.92740	+0.98184
0.11053	+1.24111	+0.59953
0.19093	+1.44380	+0.22228
0.25189	+1.48430	-0.00688
0.31316	+1.43787	-0.23885
0.39614	+1.21518	-0.63644
0.45597	+0.90901	-1.00079
0.50261	+0.57196	-1.29598
0.54416	+0.20381	-1.51178
0.58499	-0.20013	-1.61261
0.63192	-0.66631	-1.50183
0.67755	-1.04671	-1.10339
0.71649	-1.25299	-0.55731
0.75109	-1.31231	+0.01852
0.78569	-1.24515	+0.59155
0.82514	-1.02776	+1.13487
0.87125	-0.63617	+1.51860
0.91757	-0.17422	+1.61092
0.95846	+0.22851	+1.50082
1.00000	+0.59309	+1.28026

DONE

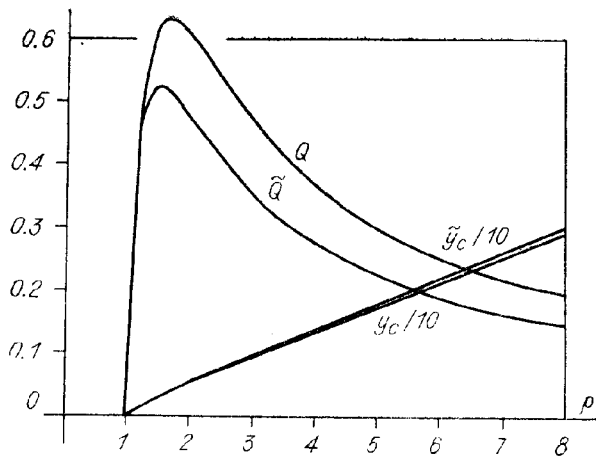


Рис. 34.

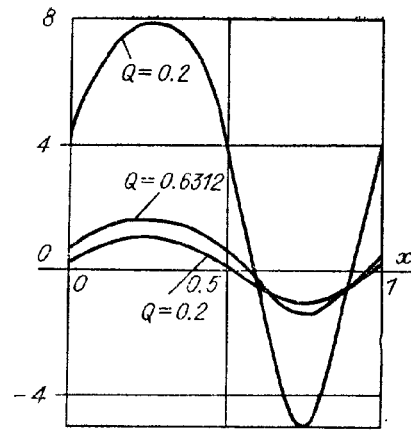


Рис. 35.

по формуле (49) (обозначение $u(x)$). Соответствие численного и приближенного решений носит очевидный характер.

Амплитуда колебаний $AMPL$ вычисляется в подпрограмме $AMPL$, обращение к которой содержится в подпрограмме $PRINT$. При задании номера j_0 обозначение $J\emptyset$ номера компоненты $y_{j_0}(x)$ вектора $y(x)$, $1 \leq j_0 \leq n$, в подпрограмме находятся ρ (обозначение $C\emptyset$), $\rho = \max_x |y_{j_0} \times(x) - y_c|$, y_c (обозначение $V\emptyset$), а также $X\emptyset$ — аргумент ρ . Для определения $V\emptyset$ и $X\emptyset$ используются сеточные значения $y_{j_0}(x)$ и $f_{j_0}(x, y, Q)$ (42). Последние вычисляются в подпрограмме $JACOB\emptyset$.

Отметим еще несколько результатов численного исследования (50) при $s_0 = \sqrt{2}$, $s_1 = s_9 = 1$. Продолжив решение по параметру, можно убедиться, что существует область изменения параметра Q , $0 < Q < Q_*$, $Q_* = 0.6312$, в которой краевая задача имеет два решения. Значение Q_* находилось с использованием подпрограммы $EXTREM$, обращение к которой описано в § 7, п. 4. Процесс вычисления Q_* отражен на рис. 33. При этом в подпрограмме $INPUT$ задаются $I2 = 1$, $Q = 0$ в стартовой позиции. Значения остальных входных параметров программы сохранены. При $Q = Q_*$ на печать выданы сеточные значения $y_1(x)$ и $y_1'(x) = y_2(x)$ (50).

На рис. 34 приведена зависимость Q от амплитуды колебаний ρ , иллюстрирующая множественность решений при $Q \geq 0$: $Q = 0$ при $\rho = 1$, $\max Q = 0.6312$ и достигается при $\rho = 1.484$. В дальнейшем с ростом $\rho > 1$ значение Q монотонно стремится к нулю. Кроме того, здесь дана зависимость $y_c(\rho)$. Приближенные значения $\tilde{Q}(\rho)$ и $\tilde{y}_c(\rho)$ получены по методу гармонического баланса (см. § 11).

Графики функций $y_1(x)$ при $Q = 0.2$ (два решения) и $Q = Q_*$ даны на рис. 35. Следует отметить явно выраженную симметрию $y_1(x)$ относительно $x = 0.25$, $0 \leq x \leq 0.5$ и симметрию $y_1(x)$ относительно $x = 0.75$, $0.5 \leq x \leq 0.75$. Еще раз обратим внимание на то обстоятельство, что построение решений краевой задачи в зависимости от Q не требует информации об их устойчивости.

§ 11. БИБЛИОГРАФИЯ И КОММЕНТАРИЙ

Метод продолжения решения по параметру рассматривался во многих работах. Сошлемся на монографии [1, 2], в которых дается обзор литературы, относящейся, в частности, к решению систем трансцендентных уравнений, содержащих параметр. Детальное описание параметризации в этом случае, использующей метод Гаусса для выбора независимых столбцов матрицы $[\Phi_y, \Phi_Q]$ (см. (2)), имеется в [3, 4]. Численному способу решения краевой задачи (1) для системы обыкновенных диф-

ференциальных уравнений, представленной в виде (9), посвящена монография [5], где приводится обзор литературы. Отметим, что продолжение решения (1) по параметру Q может быть эффективно реализовано методом ортогональных прогонок для решения линейных краевых задач на итерациях [6, 7]. Обращение к дискретной модели (10) в данном случае обусловлено использованием параметризации. В связи с вариантом адаптации сетки по x укажем на монографии [8, 9].

К идее параметризации автор обратился при решении краевой задачи (40) [10], представив задачу в виде (7), где $G(x, t)$ — теперь функция Грина краевой задачи:

$$s_0 \frac{d^4 u}{dx^4} = \frac{d^2 u}{dx^2}, \quad 0 < x < 1,$$

$$\frac{du}{dx}(0) = \frac{d^3 u}{dx^3}(0) = u(1) = \frac{du}{dx}(1) = 0.$$

Оказалось, что решение соответствующей (7) системы трансцендентных уравнений может быть найдено методом простой итерации для всех значений $s_0 \geq 0$, $0 < \lambda < 1$, где $\lambda = u(0)$ — параметр, заменивший Q . По этому поводу см. также [11—14].

Описание варианта параметризации для решения систем трансцендентных уравнений с параметром, изложенного в § 2, дано в [15, 16]. В [15] приведен текст программной реализации этого способа (программа SYSTEM).

Дискретная модель (1) в виде (10) использовалась автором независимо от [5]. Версии программы BPR-Q применялись при расчете стационарных режимов различного рода каталитических реакций. Численные примеры можно найти также в [17]. Отметим, что на проблему множественности решений в этой области математического моделирования обратил внимание автора Т. И. Зеленьяк (по этому поводу см. [18]).

Семь решений краевой задачи (37) при $\sqrt{Q} = 0.257$ приведены в [19] в виде графиков без пояснений, каким численным способом найдены решения. Таким образом, множественность решений (37), представленная на рис. 23, является вполне оригинальным результатом.

Для определения множественности решений (38) (см. [20]) можно рассмотреть задачу Коши

$$0 < x < 1,$$

$$\frac{dy_1}{dx} = y_2,$$

$$\frac{dy_2}{dx} = s_0 [(s_1 + s_2)y_1 - s_2 y_3 - Q\varphi(y_1)y_4],$$

$$\frac{dy_3}{dx} = s_2 (y_1 - y_3), \quad (51)$$

$$\frac{dy_4}{dx} = -s_3 \varphi(y_1)y_4,$$

$$y_1 = \lambda, \quad y_2 = 0, \quad y_3 = 0, \quad y_4 = 1 \quad \text{при} \quad x = 0,$$

где $\lambda > 0$, $Q > 0$ — заданные параметры. Используя численное решение (51), строится график зависимости $y_2(1)$ от λ . Число пересечений графика с осью абсцисс совпадает с числом решений (38) при фиксированном Q . Однако этот способ встречает затруднения с ростом значений параметра s_0 , так как при этом растет жесткость задачи Коши (51).

Аналогичное замечание относится и к решению задачи Коши

$$\frac{dy_1}{d\xi} = y_2 \quad (\xi > 0),$$

$$\frac{dy_2}{d\xi} = -\frac{2}{\xi} y_2 (C - y_1) \exp\left(\frac{y_1}{1 + By_1}\right),$$

$$y_1 = \lambda, \quad y_2 = 0 \quad \text{при} \quad \xi = 0,$$

где λ — заданный параметр, $0 < \lambda < C$. Отсюда можно найти $\xi = \xi_0$, при котором $y_1 = 0$, и тогда, полагая $\sqrt{Q} = \xi_0$, мы имеем решение (37), где $x = \xi/\sqrt{Q}$. Задача Коши (43) становится жесткой при $\lambda \rightarrow C$.

Параметризация может быть использована для отыскания собственного числа и собственной функции самосопряженной краевой задачи, если известно достаточно хорошее приближение собственной функции. Рассмотрим пример:

$$0 < x < 1,$$

$$\frac{dy_1}{dx} = y_2,$$

$$\frac{dy_2}{dx} = -Qy_1,$$

$$y_2(0) = y_1(1) = 0.$$

В стартовой позиции выберем в качестве текущего параметра $y_1(0)$, $y_1(0) = 1$, т. е. $\mu = 1$, $I\emptyset = 1$, $J\emptyset = 1$. Тем самым исключается возможность получить тривиальное решение. Итерационный процесс вычисления собственного числа Q и $y_1(x)$ (52) представлен на рис. 36. Здесь начальное приближение решения взято в виде $y_1(x) = 1 - x^2$, $y_2(x) = -2x$, $Q = 0$. В таблице $Y(x) = y_1(x)$, $u(x)$ — точное решение (52), $u(x) = \cos\left(\frac{\pi}{2}x\right)$, $Q\emptyset$ — точное значение наименьшего собственного числа, равного $\pi^2/4$.

То же начальное приближение использовалось в задаче

$$0 < x < 1,$$

$$\frac{dy_1}{dx} = y_2,$$

$$\frac{dy_2}{dx} = -Q(c_1 + c_2x)y_1,$$

$$y_2(0) = y_1(1) = 0$$

где $c_1 \geq 0$, $c_2 \geq 0$. На рис. 37 приведен случай $c_1 = 1$, $c_2 = 1$, а на рис. 38 — решение задачи (53) при $c_1 = 0$, $c_2 = 1$. Здесь $Y(x)$ — обозначение $y_1(x)$, $Y'(x) = y_2(x)$, $ERR(Y)$ и $ERR(Y')$ — соответствующие невязки, найденные по программе ERROR.

При исследовании вынужденных нелинейных колебаний с одной степенью свободы широко используется метод гармонического баланса. Обычно учет минимального числа слагаемых ряда Фурье, представляющего решение проблемы, оказывается достаточным для качественного описания множественности решений. Так, в рассмотренном примере (48) применение метода гармонического баланса дает следующее выражение

BPR-Q
TWO POINT BOUNDARY VALUE PROBLEM :
 $0 < X < 1$
 $Y''(X) + Q*Y(X) = 0$
 $Y'(0) = Y(1) = 0$

$$C9 = 1.2$$

$$C9 = 2.79777 E-02$$

$$C9 = 1.04594 E-04$$

$$C9 = 6.96039 E-08$$

$$C9 = 7.90805 E-08$$

$$Q = 2.4674$$

$$Q\emptyset = 2.4674$$

X	Y(X)	U(X)
0.0	1.000000	1.000000
0.1	0.987688	0.987688
0.2	0.951056	0.951056
0.3	0.891006	0.891007
0.4	0.809017	0.809017
0.5	0.707107	0.707107
0.6	0.587785	0.587786
0.7	0.453990	0.453991
0.8	0.309017	0.309018
0.9	0.156435	0.156435
1.0	-0.000000	0.000001

DONE

Рис. 36.

BPR-Q

TWO POINT BOUNDARY VALUE PROBLEM :

$$0 < X < 1$$

$$Y''(X) + Q*(1+X)*Y(X) = 0$$

$$Y'(0) = Y(1) = 0$$

$$C9 = .937495$$

$$C9 = 1.11641E-02$$

$$C9 = 2.48512E-05$$

$$C9 = 8.34006E-08$$

$$Q = 1.89597$$

X	Y(X)	Y'(X)	ERR(Y)	ERR(Y')
0.0	1.00000	+0.00000	+0.00E+00	-5.51E-07
0.1	0.99022	-0.19842	-4.29E-06	+0.00E+00
0.2	0.95983	-0.41135	-4.29E-06	-1.07E-06
0.3	0.90766	-0.63300	-4.77E-06	-1.91E-06
0.4	0.83318	-0.85614	-5.25E-06	-1.55E-06
0.5	0.73666	-1.07227	-3.81E-06	-1.91E-06
0.6	0.61928	-1.27182	-4.77E-06	-2.38E-06
0.7	0.48319	-1.44450	-4.77E-06	-3.34E-06
0.8	0.33162	-1.57981	-2.62E-06	-2.86E-06
0.9	0.16882	-1.66757	-3.46E-06	-3.34E-06
1.0	-0.00000	-1.69861	-2.98E-06	-3.76E-16

DONE

Puc. 37.

BPR-Q

TWO POINT BOUNDARY VALUE PROBLEM :

$$0 < X < 1$$

$$Y''(X) + Q*X*Y(X) = 0$$

$$Y'(0) = Y(1) = 0$$

$$C9 = 4.28567$$

$$C9 = 8.63248E-02$$

$$C9 = 7.59655E-04$$

$$C9 = 1.26687E-07$$

$$C9 = 8.41396E-09$$

$$Q = 7.83736$$

X	Y(X)	Y'(X)	ERR(Y)	ERR(Y')
0.0	1.00000	+0.00000	+0.00E+00	-3.95E-07
0.1	0.99869	-0.03917	+7.75E-06	+1.79E-07
0.2	0.98957	-0.15609	+5.72E-06	-8.94E-08
0.3	0.96498	-0.34773	+4.29E-06	-6.56E-07
0.4	0.91779	-0.60624	+1.43E-06	-2.38E-06
0.5	0.84198	-0.91697	-9.54E-07	-2.15E-06
0.6	0.73340	-1.25703	-4.77E-06	-2.15E-06
0.7	0.59064	-1.59478	-9.06E-06	-4.05E-06
0.8	0.41583	-1.89089	-1.24E-05	-3.34E-06
0.9	0.21533	-2.10098	-1.62E-05	-4.29E-06
1.0	0.00000	-2.18022	-1.79E-05	+6.78E-17

DONE

Puc. 38.

приближенного решения: $\tilde{y}(t) = \tilde{y}_c + \rho \sin(s_1 t)$,
где

$$\tilde{y}_c = \rho \sqrt{\frac{1}{2} \frac{s_0^2 - s_1^2 - s_9/\rho}{s_0^2 + s_1^2 + s_9/\rho}},$$

$$\tilde{Q} = \frac{\rho_1}{\rho} \sqrt{\frac{1}{2} \left(s_0^2 + s_1^2 - \frac{s_9}{\rho} \right) \left(s_0^2 - s_1^2 - \frac{s_9}{\rho} \right)}.$$

Зависимости $\tilde{Q}(\rho)$ и $\tilde{y}_c(\rho)$ приведены на рис. 34. Таким образом, используя программу VPR-Q, мы имеем возможность уточнить характеристики вынужденных нелинейных колебаний.

Завершим список примеров, иллюстрирующих работу программы VPR-Q, красной задачей [21]:

$$0 \leq t \leq 2\pi, \quad s_0 = 0.36, \quad s_1 \geq 0,$$

$$\frac{d^2 y}{dt^2} + s_1 \frac{dy}{dt} + y^3 = s_0 + Q \cos(t),$$

$$y(0) = y(2\pi), \quad \frac{dy}{dt}(0) = \frac{dy}{dt}(2\pi).$$

Если приближенное решение искать в виде $\tilde{y}(t) = \tilde{y}_c + a_1 \cos(t) + b_1 \sin(t)$, то согласно методу гармонического баланса зависимость \tilde{Q} от r можно представить параметрически, взяв \tilde{y}_c в качестве параметра q :

$$q = \tilde{y}_c, \quad r^2 = a_1^2 + b_1^2, \quad r^2 = \frac{2}{3q} (s_0 - q^3), \quad \tilde{Q}^2 = r^2 (z^2 + s_1^2), \quad z = 3q^2 + \frac{3}{4} r^2 - 1, \quad a_1 = \tilde{Q}z / (z^2 + s_1^2), \quad b_1 = \tilde{Q}s_1 / (z^2 + s_1^2).$$

Сопоставление при $s_1 = 0$ и $s_1 = 0.5$ амплитудной характеристики $Q(\rho)$, найденной численно по программе VPR-Q, дано на рис. 39.

Подробное описание программы VPR-Q позволяет читателю сделать ее рабочим инструментом. Очевидно, потребуются значительные усилия по воспроизводству программы и самообучению ее использования. По мнению автора, эти усилия вполне окупаются, если иметь в виду универсальный характер программы. Подборка тестовых примеров должна была, по замыслу автора, продемонстрировать это качество программы, а сопутствующие тестам выдачи на печать таблиц должны оказать помощь в отладке программы и ее применения. Как известно, версию любой программы можно улучшить, и предлагаемая тем более не является исключением. Отметим, что при написании программы на языке BASIC использовалось пособие [22].

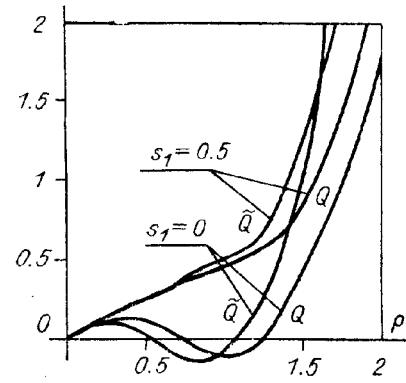


Рис. 39.

ПРИЛОЖЕНИЕ К § 6

ПРИМЕЧАНИЕ. ТЕКСТЫ НЕСТАНДАРТНЫХ ПОДПРОГРАММ BPR-Q
(INPUT, BOND, JACOB, FUNC-G, FUNC-F и PRINT)
ЗАПОЛНЕНЫ В СВЯЗИ С РЕШЕНИЕМ КРАЕВОЙ ЗАДАЧИ (35).

1. Текст программы BPR-Q (INPUT)

BPR-Q

```

100 COM D(202),V(101,2),H(101,2)
110 COM N0,M0,N,Q9,H9,I9,J9,H8,E0,N8
120 COM N9,E1,E2,E3,I2,I7,N7,D2,E4,I3
130 COM I0,I1,I6,I8,J8,K9,N1,N2,N5,N6,L5
140 GOSUB 1000
150 N1=N0+1
160 N2=2*N0
170 N5=N+1
180 N6=2*N5
190 L5=M0+1
200 I0=I1=I6=0
210 I8=I9
220 J8=J9
230 K9=1
240 CHAIN "MATRIC"
1000 REM *** SUB-PROGRAM "INPUT" ***
1010 READ N,N0,M0,Q9,H9,I9,J9,H8,E0,E4
1020 DATA I0,2,1,0,.02,12,0,1,.0001,.0001
1030 READ E1,E2,E3,N8,N9,D2,I2,I7,N7,I3
1040 DATA .001,1,1,2,5,1.1,0,1,10000,0
1050 FOR I=1 TO N+1
1060 D(I)=(I-1)/N
1070 V(I,1)=V(I,2)=0
1080 H(I,1)=H(I,2)=.1
1090 NEXT I
1100 PRINT " TWO POINT BOUNDARY VALUE PROBLEM : "
1110 PRINT "          0<X<1"
1120 PRINT "          Y''(X)+Q/[1-Y(X)]+2=0"
1130 PRINT "          Y'(0)=0 ,          Y(1)=0"
1140 PRINT
1150 PRINT "          STAPT :          Q=0, I9=N+2, J9=0"
1160 PRINT "          I2=0, N=10, N7=10000"
1170 PRINT
1180 PRINT "          Y(0)          Q          ERR(Q)
1190 PRINT "          -----
1990 RETURN
2000 END

```

2. Текст программы MATRIC

MATRIC

```
100  COM D[202],V[101,2],H[101,2]
110  COM N0,M0,N,Q9,H9,I9,J9,H8,E0,N8
120  COM N9,E1,E2,E3,I2,I7,N7,D2,E4,I3
130  COM I0,I1,I5,I8,J8,K9,N1,N2,N5,N6,L5
140  COM G[2,4],Y[202,2],P[202,2],H5,H6,H7
150  COM A[101,8],F[303,2],Q[7],Q7,P7,Q8,P8
160  DIM U[2,3],O[2],R[2],X[2,4],T[2,4],E[2,4]
1000 IF I6=N7 AND I0=0 THEN 1020
1010 GOTO 1030
1020 GOSUB 2000
1030 GOSUB 3000
1040 D=D[1]
1050 FOR K=1 TO N0
1060 Q[K]=V[1,K]
1070 NEXT K
1080 GOSUB 4000
1090 FOR K=1 TO N0
1100 X[K,1]=O[K]
1110 FOR J=1 TO N0
1120 T[K,J]=U[K,J]
1130 NEXT J
1140 X[K,3]=R[K]
1150 NEXT K
1160 FOR I=1 TO N
1170 D=D[I+1]
1180 H0=D-D[I]
1190 H1=.5*H0
1200 H2=.166667*H0
1210 H3=.125*H0
1220 FOR K=1 TO N0
1230 Q[K]=V[I+1,K]
1240 NEXT K
1250 GOSUB 4000
1260 D=.5*(D+D[I])
1270 FOR K=1 TO N0
1280 X[K,2]=O[K]
1290 FOR J=1 TO N0
1300 T[K,N0+J]=U[K,J]
1310 NEXT J
1320 X[K,4]=R[K]
1330 Q[K]=.5*(V[I,K]+Q[K])+H3*(X[K,1]-X[K,2])
1340 NEXT K
1350 GOSUB 4000
1360 FOR K=1 TO N0
1370 C0=0
1380 FOR J=1 TO N0
1390 C1=C2=0
1400 FOR M=1 TO N0
1410 C1=C1+U[K,M]*T[M,J]
1420 C2=C2+U[K,M]*T[M,N0+J]
1430 NEXT M
```

MATRIC

```

1440 E[K,J]=H2*(T[K,J]+2*U[K,J]+H1*C1)
1450 E[K,N0+J]=H2*(T[K,N0+J]+2*U[K,J]-H1*C2)
1460 C0=C0+U[K,J]*(X[J,3]-X[J,4])
1470 NEXT J
1480 E[K,K]=E[K,K]+1
1490 E[K,N0+K]=E[K,N0+K]-1
1500 F[I,K]=H2*(X[K,3]+4*R[K]+X[K,4]+H1*C0)
1510 F[I+N5,K]=V[I,K]-V[I+1,K]+H2*(X[K,1]+4*O[K]+X[K,2])
1520 NEXT K
1530 FOR K=1 TO N0
1540 K1=N2*(K-1)
1550 FOR J=1 TO N0
1560 A[I,K1+J]=E[K,J]
1570 A[I,K1+N0+J]=E[K,N0+J]
1580 T[K,J]=T[K,N0+J]
1590 NEXT J
1600 X[K,1]=X[K,2]
1610 X[K,3]=X[K,4]
1620 NEXT K
1630 NEXT I
1640 CHAIN "SOLVE"
2000 REM          *** SUB-PROGRAM "DELTA" ***
2010 IF J9=0 THEN 2030
2020 D9=D[I9]
2030 FOR I=1 TO N5
2040 D=D[I]
2050 FOR K=1 TO N0
2060 Q[K]=V[I,K]
2070 NEXT K
2080 GOSUB 4000
2090 GOSUB 5000
2100 FOR K=1 TO N0
2110 P[I,K]=O[K]
2120 C0=U[K,N0+1]
2130 FOR J=1 TO N0
2140 C0=C0+U[K,J]*O[J]
2150 NEXT J
2160 P[I+N5,K]=C0
2170 NEXT K
2180 NEXT I
2190 F[1,1]=0
2200 FOR I=1 TO N
2210 H=D[I+1]-D[I]
2220 F=.5*H*(P[I+N5,17]+P[I+1+N5,17])
2230 F[I+1,1]=F[I,1]+(12*H*ABS(P[I,17]-P[I+1,17]+F))*.25
2240 NEXT I
2250 FOR I=1 TO N-1
2260 R=F[N5,1]*I/N
2270 K=1
2280 IF R >= F[K,1] THEN 2310
2290 K=K+1
2300 GOTO 2280
2310 IF R <= F[K+1,1] THEN 2330

```

MATRIC

```

2320 GOTO 2290
2330 H=D[K+1]-D[K]
2340 T=(R-F[K,1])/(F[K+1,1]-F[K,1])
2350 D[N+2+1]=D[K]+H*T
2360 FOR J=1 TO N0
2370 A1=H*P[K,J]-V[K+1,J]+V[K,J]
2380 A2=V[K+1,J]-V[K,J]-H*P[K+1,J]
2390 Y[I+1,J]=(1-T)*V[K,J]+T*V[K+1,J]+T*
          *(1-T)*((1-T)*A1+T*A2)
2400 Y[N+2+1,J]=(1-T)*H[K,J]+T*H[K+1,J]
2410 NEXT J
2420 NEXT I
2430 FOR I=1 TO N-1
2440 D=D[I+1]=D[N+2+1]
2450 FOR K=1 TO N0
2460 Q[K]=V[I+1,K]=Y[I+1,K]
2470 H[I+1,K]=Y[N+2+1,K]
2480 NEXT K
2490 GOSUB 4000
2500 FOR K=1 TO N0
2510 P[I+1,K]=Q[K]
2520 NEXT K
2530 NEXT I
2540 I=1
2550 D0=D[I+1]-D[I]
2560 IF D0<D2 THEN 2820
2570 FOR K=1 TO N0
2580 Y[I,K]=D0*P[I,K]-V[I+1,K]+V[I,K]
2590 Y[N+2+1,K]=V[I+1,K]-V[I,K]-D0*P[I+1,K]
2600 NEXT K
2610 K1=INT(D0/D2)
2620 FOR K=N+1 TO I+1 STEP -1
2630 D[K+K1]=D[K]
2640 FOR J=1 TO N0
2650 V[K+K1,J]=V[K,J]
2660 H[K+K1,J]=H[K,J]
2670 P[K+K1,J]=P[K,J]
2680 NEXT J
2690 NEXT K
2700 H=D0/(K1+1)
2710 FOR K=1 TO K1
2720 D[I+K]=D[I]+H*K
2730 T=K/(K1+1)
2740 FOR J=1 TO N0
2750 V=(1-T)*Y[I,J]+T*Y[N+2+1,J]
2760 V[I+K,J]=(1-T)*V[I,J]+T*V[I+1+K1,J]+T*(1-T)*V
2770 H[I+K,J]=(1-T)*H[I,J]+T*H[I+1+K1,J]
2780 NEXT J
2790 NEXT K
2800 N=N+K1
2810 I=I+K1
2820 I=I+1
2830 IF I>N THEN 2850

```

MATRIC

```

2840 GOTO 2550
2850 IF I9=1 THEN 2960
2860 IF J9>0 THEN 2890
2870 I9=N+2
2880 GOTO 2960
2890 I=1
2900 IF D9 >= D[I] THEN 2930
2910 I=I+1
2920 GOTO 2900
2930 IF D9 <= D[I+1] THEN 2950
2940 GOTO 2910
2950 I9=I+(D9>.5*(D[I+1]+D[I]))
2960 N5=N+1
2970 N6=2*N5
2990 RETURN
3000 REM          *** SUB-PROGRAM "BOND" ***
3010 G[1,1]=G[1,3]=G[2,2]=G[2,3]=0
3020 G[1,2]=G[2,1]=1
3030 G[1,4]=V[1,2]
3040 G[2,4]=V[N5,1]
3990 RETURN
4000 REM          *** SUB-PROGRAM "JACOB" ***
4010 U[1,1]=R[1]=U[2,2]=0
4020 U[1,2]=1
4030 O[1]=Q[2]
4040 IF ABS(1-Q[1])>.000001 THEN 4060
4050 Q[1]=.99999
4060 R[2]=-1/(1-Q[1])+2
4070 O[2]=Q9*R[2]
4080 U[2,1]=2*O[2]/ABS(1-Q[1])
4990 RETURN
5000 REM          *** SUB-PROGRAM "ADAPT" ***
5010 U[1,3]=U[2,3]=0
5990 RETURN
6000 END

```

3. Текст программы SOLVE

SOLVE

```

100 COM D[202],V[101,2],H[101,2]
110 COM N0,M0,N,Q9,H9,I9,J9,H8,E0,N8
120 COM N9,E1,E2,E3,I2,I7,N7,D2,E4,I3
130 COM I0,I1,I6,I8,J8,K9,N1,N2,N5,N6,L5
140 COM G[2,4],Y[202,2],P[202,2],H5,H6,H7
150 COM A[101,8],F[303,2],Q[7],Q7,P7,Q8,P8
160 DIM W[4,7]
1000 L0=2+(I9<N+2)
1010 FOR K=1 TO M0
1020 FOR J=1 TO N0
1030 W[K,J]=G[K,J]
1040 NEXT J
1050 W[K,N2+1]=G[K,N0+1]
1060 W[K,N2+2]=G[K,N0+2]
1070 W[K,N2+3]=0
1080 NEXT K
1090 IF L0=2 THEN 1420
1100 FOR I=1 TO N
1110 FOR K=1 TO N0
1120 F[I+N6,K]=0
1130 NEXT K
1140 NEXT I
1150 J0=J9+1
1160 IF I9>1 THEN 1240
1170 FOR K=1 TO M0
1180 FOR J=J0 TO N0
1190 W[K,J-1]=W[K,J]
1200 NEXT J
1210 W[K,N0]=0
1220 W[K,N2+3]=G[K,J9]
1230 NEXT K
1240 IF I9=N5 THEN 1340
1250 FOR K=1 TO N0
1260 K1=N2*(K-1)
1270 F[N6+I9,K]=A[I9,K1+J9]
1280 FOR J=J0 TO N2
1290 A[I9,K1+J-1]=A[I9,K1+J]
1300 NEXT J
1310 A[I9,K1+N2]=0
1320 NEXT K
1330 IF I9=1 THEN 1420
1340 FOR K=1 TO N0
1350 K2=N2*(K-1)+N0
1360 F[I9-1+N6,K]=A[I9-1,K2+J9]
1370 FOR J=J0 TO N0
1380 A[I9-1,K2+J-1]=A[I9-1,K2+J]
1390 NEXT J
1400 A[I9-1,K2+N0]=0
1410 NEXT K
1420 N3=N2+L0
1430 FOR I=1 TO N
1440 M8=M0+(I>I9)

```


SOLVE

```
1450 FOR K=1 TO M8
1460 FOR J=N1 TO N2
1470 W[K,J]=0
1480 NEXT J
1490 NEXT K
1500 FOR K=1 TO N0
1510 K1=N2*(K-1)
1520 FOR J=1 TO N2
1530 W[K+M8,J]=A[I,K1+J]
1540 NEXT J
1550 FOR M=1 TO L0
1560 W[K+M8,N2+M]=F[I+N5*(M-1),K]
1570 NEXT M
1580 NEXT K
1590 L1=N0-(I=19)
1600 N4=N0+M8
1610 GOSUB 2670
1620 FOR K=1 TO L1
1630 K1=N2*(K-1)
1640 FOR J=1 TO N2
1650 A[I,K1+J]=W[K,J]
1660 NEXT J
1670 FOR M=1 TO L0
1680 F[I+N5*(M-1),K]=W[K,N2+M]
1690 NEXT M
1700 IF L0=3 THEN 1720
1710 F[I+N6,K]=F[I,K]
1720 NEXT K
1730 M9=L5-(I<19)
1740 FOR K=1 TO M9
1750 FOR J=1 TO N0
1760 W[K,J]=W[L1+K,L1+J]
1770 NEXT J
1780 FOR M=1 TO L0
1790 W[K,N2+M]=W[L1+K,N2+M]
1800 NEXT M
1810 NEXT K
1820 NEXT I
1830 L2=(I9<N5)
1840 N4=N0+L2
1850 L4=N0-(I9=N5)
1860 FOR K=L5 TO N0
1870 FOR J=1 TO N0
1880 W[K+L2,J]=G[K,J]
1890 NEXT J
1900 W[K+L2,N2+1]=G[K,N0+1]
1910 W[K+L2,N2+2]=G[K,N0+2]
1920 W[K+L2,N2+3]=0
1930 IF L4=N0 THEN 1990
1940 FOR J=J0 TO N0
1950 W[K,J-1]=W[K,J]
1960 NEXT J
```

SOLVE

```

1970 W[K,N0]=0
1980 W[K,N2+3]=G[K,J9]
1990 NEXT K
2000 FOR K=1 TO N4
2010 FOR M=1 TO L0
2020 W[K,N0+M]=W[K,N2+M]
2030 NEXT M
2040 IF L0=3 THEN 2060
2050 W[K,N0+3]=W[K,N0+1]
2060 NEXT K
2070 L1=L2+L4
2080 N3=N0+3
2090 GOSUB 2670
2100 FOR K=1 TO N0
2110 K1=N2*(K-1)
2120 FOR J=1 TO N0
2130 A[N5,K1+J]=W[K,J]
2140 NEXT J
2150 FOR M=1 TO 3
2160 F[N5*M,K]=W[K,N0+M]
2170 NEXT M
2180 NEXT K
2190 FOR M=1 TO K9
2200 FOR K=1 TO N0
2210 Q[K]=0
2220 NEXT K
2230 Q[N2+M]=-(M=2)
2240 IF L0=2 THEN 2330
2250 Q[N2+M]=W[N4,N1+M]/W[N4,N1]
2260 FOR I=1 TO N5
2270 M7=I+N5*M
2280 L1=N0-(I=19)
2290 FOR K=1 TO L1
2300 F[M7,K]=F[M7,K]-Q[N2+M]*F[I,K]
2310 NEXT K
2320 NEXT I
2330 FOR I=N5 TO 1 STEP -1
2340 L1=N0-(I=19)
2350 L2=N2*(L1-1)+L1
2360 M7=I+N5*M
2370 F[M7,L1]=(F[M7,L1]-Q[L1])/A[I,L2]
2380 K0=L1-1
2390 FOR K=K0 TO 1 STEP -1
2400 C0=F[M7,K]-Q[K]
2410 K1=N2*(K-1)
2420 K2=K+1
2430 FOR J=L1 TO K2 STEP -1
2440 C0=C0-A[I,K1+J]*F[M7,J]
2450 NEXT J
2460 F[M7,K]=C0/A[I,K1+K]
2470 NEXT K
2480 IF I=1 THEN 2580

```

SOLVE

```
2490 L3=N0-(I=I9+1)
2500 FOR K=1 TO L3
2510 K2=N2*(K-1)+L3
2520 C0=0
2530 FOR J=1 TO L1
2540 C0=C0+A[I-1,K2+J]*F[M7,J]
2550 NEXT J
2560 Q[K]=C0
2570 NEXT K
2580 NEXT I
2590 IF L0=2 THEN 2650
2600 M7=I9+N5*M
2610 FOR K=N0 TO J0 STEP -1
2620 F[M7,K]=F[M7,K-1]
2630 NEXT K
2640 F[M7,J9]=-(M=2)
2650 NEXT M
2660 CHAIN "NEWTON"
2670 REM *** SUB-PROGRAM "GAUSS" ***
2680 FOR M=1 TO L1
2690 C0=0
2700 FOR K=M TO N4
2710 IF ABS(W[K,M])<ABS(C0) THEN 2740
2720 C0=W[K,M]
2730 K0=K
2740 NEXT K
2750 FOR J=M TO N3
2760 Q[J]=W[K0,J]
2770 W[K0,J]=W[M,J]
2780 W[M,J]=Q[J]
2790 NEXT J
2800 K1=M+1
2810 FOR K=K1 TO N4
2820 R=W[K,M]/C0
2830 FOR J=K1 TO N3
2840 W[K,J]=W[K,J]-R*Q[J]
2850 NEXT J
2860 NEXT K
2870 NEXT M
2880 RETURN
2890 END
```

4. Текст программы NEWTON

NEWTON

```

100 COM D[202],V[101,2],H[101,2]
110 COM N0,M0,N,Q9,H9,I9,J9,H8,E0,N8
120 COM N9,E1,E2,E3,I2,I7,N7,D2,E4,I3
130 COM I0,I1,I6,I8,J8,K9,N1,N2,N5,N6,L5
140 COM G[2,4],Y[202,2],P[202,2],H5,H6,H7
150 COM A[101,8],F[303,2],Q[7],Q7,P7,Q8,P8
1000 C9=ABS(Q[N2+1])
1010 C0=ABS(Q9)
1020 FOR I=1 TO N5
1030 FOR K=1 TO N0
1040 C9=(C9 MAX ABS(F[I+N5,K]))
1050 C0=(C0 MAX ABS(V[I,K]))
1060 V[I,K]=V[I,K]-F[I+N5,K]
1070 NEXT K
1080 NEXT I
1090 Q9=Q9-Q[N2+1]
1100 C9=C9/(C0+(C0=0))
1110 IF C9<E0 AND K9=2 THEN 1470
1120 I0=I0+1
1130 IF C9>E0 THEN 1160
1140 K9=2
1150 CHAIN "MATRIC"
1160 K9=1
1170 IF I0>N8 THEN 1190
1180 CHAIN "MATRIC"
1190 IF I1>0 THEN 1230
1200 IF I0<N9 THEN 1180
1210 PRINT "I1=0, I0=M9, STOP"
1220 STOP
1230 H7=H7/2
1240 IF ABS(H7)>E1*H5 THEN 1270
1250 PRINT "I1=";I1,"ABS(H7)<E1*H5, STOP"
1260 STOP
1270 I0=0
1280 IF I1>1 THEN 1360
1290 FOR I=1 TO N5
1300 FOR K=1 TO N0
1310 V[I,K]=Y[I+N5,K]+H7*P[I+N5,K]
1320 NEXT K
1330 NEXT I
1340 Q9=Q8+H7*P8
1350 CHAIN "MATRIC"
1360 FOR I=1 TO N5
1370 FOR K=1 TO N0
1380 F1=P[I+N5,K]+P[I,K]-2*(Y[I+N5,K]-Y[I,K])/H6
1390 F2=F1+P[I+N5,K]-(Y[I+N5,K]-Y[I,K])/H6
1400 V[I,K]=Y[I+N5,K]+H7*(P[I+N5,K]+(H7/H6)*
                                *(F2+(H7/H6)*F1))
1410 NEXT K
1420 NEXT I
1430 F1=P8+P7-2*(Q8-Q7)/H6
1440 F2=F1+P8-(Q8-Q7)/H6
1450 Q9=Q8+H7*(P8+(H7/H6)*(F2+(H7/H6)*F1))
1460 CHAIN "MATRIC"

```

NEWTON

```

1470 K9=1
1480 C0=0
1490 FOR I=1 TO N5
1500 FOR K=1 TO N0
1510 F[I+N6,K]=-F[I+N6,K]
1520 IF I2>N0 THEN 1570
1530 IF ABS(F[I+N6,K])<ABS(C0) THEN 1570
1540 I9=I
1550 J9=K
1560 C0=F[I+N6,K]
1570 NEXT K
1580 NEXT I
1590 Q[N2+2]=-Q[N2+2]
1600 IF I2>0 THEN 1660
1610 IF ABS(Q[N2+2])<ABS(C0) THEN 1770
1620 I9=N+2
1630 J9=0
1640 C0=Q[N2+2]
1650 GOTO 1770
1660 IF I2=N1 THEN 1620
1670 IF I2>N1 THEN 1760
1680 J9=I2
1690 C0=0
1700 FOR I=1 TO N5
1710 IF ABS(F[I+N6,J9])<ABS(C0) THEN 1740
1720 I9=I
1730 C0=F[I+N6,J9]
1740 NEXT I
1750 GOTO 1770
1760 C0=F[I9+N6,J9]
1770 FOR I=1 TO N5
1780 FOR K=1 TO N0
1790 F[I+N6,K]=F[I+N6,K]/C0
1800 NEXT K
1810 NEXT I
1820 Q[N2+2]=Q[N2+2]/C0
1830 IF I3=0 THEN 1850
1840 CHAIN "ERROR"
1850 GOSUB 3000
1860 IF I1>0 THEN 2070
1870 IF I6=N7 THEN 1920
1880 IF I9=N+2 THEN 1910
1890 H7=H5=H[I9,J9]
1900 GOTO 1920
1910 H7=H5=H9
1920 IF J8=0 THEN 1950
1930 P=F[I8+N6,J8]*H7
1940 GOTO 1960
1950 P=Q[N2+2]*H7
1960 IF SGN(P)=H8 THEN 1980
1970 H7=-H7

```

NEWTON

```

1980 FOR I=1 TO N5
1990 FOR K=1 TO N0
2000 Y[I+N5,K]=V[I,K]
2010 P[I+N5,K]=F[I+N6,K]
2020 NEXT K
2030 NEXT I
2040 Q8=Q9
2050 P8=Q[N2+2]
2060 GOTO 2310
2070 IF I9=N+2 THEN 2120
2080 P0=P[I9+N5,J9]
2090 H5=H[I9,J9]
2100 H6=V[I9,J9]-Y[I9+N5,J9]
2110 GOTO 2150
2120 P0=P8
2130 H5=H9
2140 H6=Q9-Q8
2150 FOR I=1 TO N5
2160 FOR K=1 TO N0
2170 Y[I,K]=Y[I+N5,K]
2180 P[I,K]=P[I+N5,K]/P0
2190 Y[I+N5,K]=V[I,K]
2200 P[I+N5,K]=F[I+N6,K]
2210 NEXT K
2220 NEXT I
2230 Q7=Q8
2240 P7=P8/P0
2250 Q8=Q9
2260 P8=Q[N2+2]
2270 H7=E2*H6
2280 IF ABS(H7)<E3*H5 THEN 2300
2290 H7=E3*H5*SGN(H6)
2300 IF I1 >= N7 THEN 2340
2310 I1=I1+1
2320 I6=0
2330 GOTO 1270
2340 C0=0
2350 FOR I=1 TO N5 STEP N
2360 FOR K=1 TO N0
2370 IF ABS(P[I+N5,K])<C0 THEN 2410
2380 I8=I
2390 J8=K
2400 C0=ABS(P[I+N5,K])
2410 NEXT K
2420 NEXT I
2430 H8=SGN(H7*P[I8+N5,J8])
2440 IF ABS(P8)<C0 THEN 2490
2450 I8=N+2
2460 J8=0
2470 C0=ABS(P8)
2480 H8=SGN(H7*P8)
2490 IF C0<E4 THEN 2310

```

NEWTON

```
2500 I6=N7
2510 I0=I1=0
2520 CHAIN "MATRIC"
3000 REM *** SUB-PROGRAM "PRINT" ***
3010 S0=V[1,1]
3020 IF S0 >= .999999 THEN 3090
3030 S1=SQR(S0)
3040 S=S1+(1-S0)*LOG((1+S1)/SQR(1-S0))
3050 Q0=.5*(1-S0)*S+2
3060 PRINT USING "10X,2(D.5D,5X),SD.3DE,5X,DD,5X,DD";
                                     S0,Q0,Q0-Q0,I9,J9
3070 IF Q0<.001 AND Q0>0 THEN 3090
3080 GOTO 3990
3090 STOP
3990 RETURN
4000 END
```

5. Текст программы ERROR

ERROR

```
100 COM D[202],V[101,2],H[101,2]
110 COM N0,M0,N,Q9,H9,I9,J9,H8,E0,N8
120 COM N9,E1,E2,E3,I2,I7,N7,D2,E4,I3
130 COM I0,I1,I6,I8,J8,K9,N1,N2,N5,N6,L5
140 COM G[2,4],Y[202,2],P[202,2],H5,H6,H7
150 COM A[101,8],F[303,2],Q[7],Q7,P7,Q8,P8
160 DIM U[2,3],X[2,2]
1000 REM *** SUB-PROGRAM "ERROR" ***
1010 GOSUB 2000
1020 FOR I=1 TO N5
1030 D=D[I]
1040 FOR K=1 TO N0
1050 Q[K]=V[I,K]
1060 NEXT K
1070 GOSUB 3000
1080 FOR K=1 TO N0
1090 F[I+N5,K]=Q[K]
1100 NEXT K
1110 NEXT I
1120 FOR I=1 TO N
1130 H=D[I+1]-D[I]
1140 FOR K=1 TO N0
1150 X[K,1]=V[I,K]-V[I+1,K]+H*F[I+N5,K]
1160 X[K,2]=V[I+1,K]-V[I,K]-H*F[I+1+N5,K]
1170 NEXT K
1180 FOR M=1 TO 3
1190 T=.25*M
1200 D=D[I]+H*T
```

ERROR

```

1210 FOR K=1 TO N0
1220 C0=(1-T)*X[K,1]+T*X[K,2]
1230 Q[K]=(1-T)*V[I,K]+T*V[I+1,K]+T*(1-T)*C0
1240 NEXT K
1250 GOSUB 3000
1260 FOR K=1 TO N0
1270 U[K,M]=O[K]
1280 NEXT K
1290 NEXT M
1300 FOR K=1 TO N0
1310 C1=7*(F[I+N5,K]+F[I+1+N5,K])+32*(U[K,1]+U[K,3])
1320 F[I+1,K]=(V[I+1,K]-V[I,K])/H-(C1+12*U[K,2])/90
1330 NEXT K
1340 NEXT I
1350 FOR K=1 TO M0
1360 F[1,K]=G[K,N0+2]
1370 NEXT K
1380 FOR I=1 TO N
1390 FOR K=M0+1 TO N0
1400 F[I,K]=F[I+1,K-M0]
1410 NEXT K
1420 FOR K=1 TO M0
1430 F[I+1,K]=F[I+1,N0-M0+K]
1440 NEXT K
1450 NEXT I
1460 FOR K=M0+1 TO N0
1470 F[N5,K]=G[K,N0+2]
1480 NEXT K
1490 CHAIN "NEWTON",1850
2000 REM *** SUB-PROGRAM "FUNC-G" ***
2010 G[1,4]=V[1,2]
2020 G[2,4]=V[N5,1]
2990 RETURN
3000 REM *** SUB-PROGRAM "FUNC-F" ***
3010 O[1]=Q[2]
3020 IF ABS(1-Q[1])>.000001 THEN 3040
3030 Q[1]=.99999
3040 O[2]=-Q9/(1-Q[1])*2
3990 RETURN
4000 END

```


ПРИМЕЧАНИЕ. ЗДЕСЬ ПО-ПРЕЖНЕМУ РАССМАТРИВАЕТСЯ РЕШЕНИЕ КОНКРЕТНОЙ КРАЕВОЙ ЗАДАЧИ. ОДНАКО, ПРАВИЛО ОБРАЩЕНИЯ К СТАНДАРТНЫМ ПОДПРОГРАММАМ ADAPT, ERROR, SOLV-Q и EXTREM НОСЯТ ОБЩИЙ ХАРАКТЕР.

1. Тексты подпрограмм INPUT и PRINT, содержащие пример обращения к подпрограммам ADAPT и ERROR

BPR-Q

```

1000 REM *** SUB-PROGRAM "INPUT" ***
1010 READ N,N0,M0,Q9,H9,I9,J9,H8,E0,E4
1020 DATA 10,2,1,.15,0,1,1,1,.0001,.0001
1030 READ E1,E2,E3,N8,N9,D2,I2,I7,N7,I3
1040 DATA .001,1,1,2,10,1.1,4,1,0,1
1050 Y0=.8
1060 FOR I=1 TO N+1
1070 D[I]=(I-1)/N
1080 V[I,1]=Y0*(1-D[I]^2)
1090 V[I,2]=-2*Y0*(1-D[I])
1100 H[I,1]=H[I,2]=0
1110 NEXT I
1120 PRINT "TWO POINT BOUNDARY VALUE PROBLEM : "
1130 PRINT " 0<X<1"
1140 PRINT " Y''(X)+Q/[1-Y(X)]^2=0"
1150 PRINT " Y'(0)=0, Y(1)=0"
1160 PRINT
1170 PRINT "START : Q=.15, I9=1, J9=1, Y(X)=.8*(1-X^2)"
1180 PRINT " I2=4, N=10, N7=0, D2=1.1, I3=1, I6=-1"
1190 PRINT
1990 RETURN
2000 END
    
```

NEWTON

```

3000 REM *** SUB-PROGRAM "PRINT" ***
3010 S0=V[I,1]
3020 S1=SQR(S0)
3030 S=S1+(1-S0)*LOG((1+S1)/SQR(1-S0))
3040 Q0=.5*(1-S0)*S^2
3050 PRINT " Q="; Q9; "ERR(Q)="; Q9-Q0
3060 PRINT
3070 PRINT " X Y(X) Y'(X) ERR(Y) "
3080 PRINT " -----"
3090 FOR I=1 TO N+1
3100 PRINT USING 3110; D[I],V[I,1],V[I,2],F[I,1],F[I,2]
3110 IMAGE 2X,2(D.5D,3X),SDD.5D,2(3X,SD.3DE)
3120 NEXT I
3130 PRINT
3140 IF I6<0 THEN 3990
3150 STOP
3990 RETURN
4000 END
    
```

2. Текст подпрограммы PRINT с обращением к подпрограмме SOLV-Q

NEWTON

```

3000 REM *** SUB-PROGRAM "PRINT" ***
3010 PRINT USING 3020; V[1, 1], Q9, Q[N2+2]
3020 IMAGE 5X, "Y(0)=", DD. 5D, 5X, "Q=", DD. 5D, 5X, "Q'=", DD. 5D
3030 IF V[1, 1] <= .9 THEN 3050
3040 STOP
3050 E5=.0001
3060 Q0=.2
3070 GOSUB 5000
3080 IF P>0 THEN 3990
3090 PRINT
3100 PRINT "          X          Y(X)          Y'(X)          "
3110 PRINT "          -----"
3120 FOR I=1 TO N+1
3130 PRINT USING 3140; D(I), V[I, 1], V[I, 2]
3140 IMAGE 7X, 2(D. 5D, 3X), SDD. 5D
3150 NEXT I
3160 PRINT
3170 STOP
3990 RETURN

```

NEWTON

```

5000 REM *** SUB-PROGRAM "SOLV-Q" ***
5010 IF I1=0 THEN 5320
5020 P0=P[I9+N5, J9]
5030 FOR I=1 TO N5
5040 FOR J=1 TO N0
5050 P[I+N5, J]=P[I+N5, J]/P0
5060 NEXT J
5070 NEXT I
5080 P8=P8/P0
5090 H=V[I9, J9]-Y[I9+N5, J9]
5100 H1=(Q9-Q0)/(Q0+(Q0=0))
5110 IF ABS(H)<E5*H5 OR ABS(H1)<E5 THEN 5340
5120 IF SGN(Q0-Q9)=SGN(Q0-Q8) THEN 5320
5130 A=Q8-Q9+H*P8
5140 B=Q9-Q8-H*Q[N2+2]
5150 R1=Q8
5160 R2=Q9
5170 T1=0
5180 T2=1
5190 IF T2-T1<.0001 OR T2<T1 THEN 5290
5200 T=.5*(T1+T2)
5210 Q=(1-T)*Q8+T*Q9+T*(1-T)*((1-T)*A+T*B)
5220 IF SGN(Q-Q0)=SGN(R2-Q0) THEN 5260
5230 R1=Q
5240 T1=T
5250 GOTO 5190
5260 R2=Q
5270 T2=T
5280 GOTO 5190
5290 E2=T-1
5300 GOTO 5320

```

NEWTON

```
5310 E2=1
5320 P=1
5330 GOTO 5990
5340 P=-1
5990 RETURN
```

3. Текст подпрограммы PRINT с обращением к подпрограмме EXTREM

NEWTON

```
3000 REM *** SUB-PROGRAM "PRINT" ***
3010 PRINT USING 3020; V[1,1], Q9, Q[N2+2]
3020 IMAGE 5X, "Y(0)=", DD.5D, 5X, "Q=", DD.5D, 5X, "Q'=", SD.5D
3030 IF V[1,1] <= .9 THEN 3050
3040 STOP
3050 E5=.0001
3060 GOSUB 4000
3070 IF P>0 THEN 3990
3080 PRINT
3090 PRINT " X Y(X) Y'(X) "
3100 PRINT " -----"
3110 FOR I=1 TO N+1
3120 PRINT USING 3130; D[I], V[I,1], V[I,2]
3130 IMAGE 7X, 2(D.5D, 3X), SDD.5D
3140 NEXT I
3150 PRINT
3160 STOP
3990 RETURN
4000 REM *** SUB-PROGRAM "EXTREM" ***
4010 IF I1=0 THEN 4220
4020 P0=P[I9+N5, J9]
4030 FOR I=1 TO N5
4040 FOR J=1 TO N0
4050 P[I+N5, J]=P[I+N5, J]/P0
4060 NEXT J
4070 NEXT I
4080 P8=P8/P0
4090 H=V[I9, J9]-Y[I9+N5, J9]
4100 IF ABS(H)<E1*H5 OR ABS(Q[N2+2])<E5 THEN 4240
4110 IF SGN(Q[N2+2])=SGN(P8) THEN 4210
4120 H0=H*P8
4130 A=Q8-Q9+H*P8
4140 B=Q9-Q8-H*Q[N2+2]
4150 C=2*A-B
4160 T=H0/(C+SQR(ABS(C^2+3*(B-A)*H0)))
4170 IF T>0 AND T<1 THEN 4190
4180 T=H0/(3*(A-B)*T)
4190 E2=T-1
4200 GOTO 4220
4210 E2=1
4220 P=1
4230 GOTO 4990
4240 P=-1
4990 RETURN
```

ПРИЛОЖЕНИЕ К § 8

В ПРИЛОЖЕНИИ ПРИВЕДЕНЫ ПРИМЕРЫ ЗАПОЛНЕНИЯ
НЕСТАНДАРТНЫХ ПОДПРОГРАММ ПРОГРАММЫ BPR-Q.
ФОРМУЛИРОВКИ СООТВЕТСТВУЮЩИХ КРАЕВЫХ ЗАДАЧ
СОДЕРЖАТСЯ В ТЕКСТАХ ПОДПРОГРАММЫ INPUT

Пример 1

BPR-Q

```

1000 REM *** SUB-PROGRAM "INPUT" ***
1010 READ N,N0,M0,Q9,H9,I9,J9,H8,E0,E4
1020 DATA 10,2,1,0,.02,1,1,1,.0001,.0001
1030 READ E1,E2,E3,N8,N9,D2,I2,I7,N7,I3
1040 DATA .001,1,1,5,10,.2,1,1,3,0
1050 FOR I=1 TO N+1
1060 D[I]=(I-1)/N
1070 V[I,1]=V[I,2]=0
1080 H[I,1]=H[I,2]=.5
1090 NEXT I
1100 PRINT " TWO POINT BOUNDARY VALUE PROBLEM : "
1110 PRINT " 0<X<1, C=30, B=1/60"
1120 PRINT " Y''+2*Y'/X + Q*(C-Y)*EXP(Y/(1+B*Y)) = 0"
1130 PRINT " Y'(0)=0, Y(1)=0"
1140 PRINT
1150 PRINT " Y(0) SQR(Q) LOG(FACTOR) Y'(1)
1160 PRINT " -----
1170 PRINT
1990 RETURN
2000 END

```

MATRIC

```

3000 REM *** SUB-PROGRAM "BOND" ***
3010 G[1,1]=G[1,3]=G[2,2]=G[2,3]=0
3020 G[1,2]=G[2,1]=1
3030 G[1,4]=V[1,2]
3040 G[2,4]=V[N5,1]
3990 RETURN
4000 REM *** SUB-PROGRAM "JACOB" ***
4010 C=30
4020 B=1/60
4030 Q[1]=Q[2]
4040 U[1,1]=R[1]=0
4050 U[1,2]=1
4060 Z0=1/(1+B*Q[1])
4070 E=EXP(Z0*Q[1])
4080 IF ABS(D)<.000001 THEN 4140
4090 R[2]=-(C-Q[1])*E
4100 U[2,2]=-2/D

```

MATRIC

```

4110  O[2]=Q9*R[2]+U[2,2]*Q[2]
4120  U[2,1]=Q9*(1-(C-Q[1])*Z0^2)*E
4130  GOTO 4990
4140  R[2]=- (C-Q[1])*E/3
4150  U[2,2]=0
4160  O[2]=Q9*R[2]
4170  U[2,1]=Q9*(1-(C-Q[1])*Z0^2)*E/3
4990  RETURN
5000  REM          *** SUB-PROGRAM "ADAPT" ***
5010  U[1,3]=0
5020  IF ABS(D)<.000001 THEN 5050
5030  U[2,3]=2/D^2
5040  GOTO 5990
5050  U[2,3]=0
5990  RETURN
6000  END

```

NEWTON

```

3000  REM          *** SUB-PROGRAM "PRINT" ***
3010  C=30
3020  B=1/60
3030  IF Q9>.000001 THEN 3060
3040  F=0
3050  GOTO 3070
3060  F=LOG(-3*V[NS,2]/C/Q9)/LOG(10)
3070  PRINT USING 3080;V[1,1],SQR(Q9),F,V[NS,2],Q[N2+2]
3080  IMAGE 4X,DD,D,4X,D.5D,4X,D.5D,4X,SDD.5D,4X,SD.5DE
3090  IF V[1,1] <= 15 THEN 3990
3100  STOP
3990  RETURN

```

ERROR

```

2000  REM          *** SUB-PROGRAM "FUNC-G" ***
2010  G[1,4]=V[1,2]
2020  G[2,4]=V[NS,1]
2990  RETURN
3000  REM          *** SUB-PROGRAM "FUNC-F" ***
3010  C=30
3020  B=1/60
3030  O[1]=Q[2]
3040  IF ABS(D)<.000001 THEN 3070
3050  O[2]=Q9*(C-Q[1])*EXP(Q[1]/(1+B*Q[1]))
3060  GOTO 3990
3070  O[2]=Q9*(C-Q[1])*EXP(Q[1]/(1+B*Q[1]))/3
3990  RETURN
4000  END

```

Пример 2

BPR-Q

```

1000  REM                      *** SUB-PROGRAM "INPUT" ***
1010  READ N,N0,M0,Q9,H9,I9,J9,H8,E0,E4
1020  DATA 5,4,3,0,10,6,1,1,.0001,.0001
1030  READ E1,E2,E3,N8,N9,D2,I2,I7,N7,I3
1040  DATA .001,1.25,1,2,7,.2,1,1,3,0
1050  FOR I=1 TO N+1
1060  D[I]=(I-1)/N
1070  V[I,1]=V[I,2]=V[I,3]=0
1080  V[I,4]=1
1090  H[I,1]=H[I,2]=H[I,3]=H[I,4]=.25
1100  NEXT I
1110  PRINT "                TWO POINT BOUNDARY VALUE PROBLEM : "
1120  PRINT "                                0<X<1"
1130  PRINT "                                Y' = V"
1140  PRINT "                                V' = S0*((S1+S2)*Y-S2*W-Q*(Y+S4)*Z)"
1150  PRINT "                                W' = S2*(Y-W)"
1160  PRINT "                                Z' = -S3*(Y+S4)*Z"
1170  PRINT "                                X=0 : V=W=0, Z=1"
1180  PRINT "                                X=1 : V=0"
1190  PRINT "                                F(U) = 1/[S5+EXP(-U/(1+S6*U))]"
1200  PRINT
1210  PRINT " S0=10, S1=8, S2=5, S3=5, S4=-4, S5=.5, S6=.05"
1220  PRINT
1990  RETURN
2000  END

```

MATRIC

```

3000  REM                      *** SUB-PROGRAM "BOND" ***
3010  G[1,1]=G[1,3]=G[1,4]=G[1,5]=0
3020  G[2,1]=G[2,2]=G[2,4]=G[2,5]=0
3030  G[3,1]=G[3,2]=G[3,3]=G[3,5]=0
3040  G[4,1]=G[4,3]=G[4,4]=G[4,5]=0
3050  G[1,2]=G[2,3]=G[3,4]=G[4,2]=1
3060  G[1,6]=V[1,2]
3070  G[2,6]=V[1,3]
3080  G[3,6]=V[1,4]-1
3090  G[4,6]=V[N5,2]
3990  RETURN

```

MATRIC

```

4000 REM *** SUB-PROGRAM "JACOB" ***
4010 READ S0,S1,S2,S3,S4,S5,S6
4020 DATA 10,8,5,5,-4,.5,.05
4030 RESTORE 4010
4040 Z0=1/(1+S6*(S4+Q[1]))
4050 E=EXP(-(S4+Q[1])*Z0)
4060 F=1/(S5+E)
4070 P=E*(F*Z0)+2
4080 Q[1]=Q[2]
4090 U[1,1]=U[1,3]=U[1,4]=R[1]=0
4100 U[1,2]=1
4110 Q[2]=S0*((S1+S2)*Q[1]-S2*Q[3]-Q9*F*Q[4])
4120 U[2,1]=S0*(S1+S2-Q9*P*Q[4])
4130 U[2,2]=0
4140 U[2,3]=-S0*S2
4150 U[2,4]=-S0*Q9*F
4160 R[2]=-S0*F*Q[4]
4170 Q[3]=S2*(Q[1]-Q[3])
4180 U[3,1]=S2
4190 U[3,2]=U[3,4]=R[3]=0
4200 U[3,3]=-S2
4210 Q[4]=-S3*F*Q[4]
4220 U[4,1]=-S3*P*Q[4]
4230 U[4,2]=U[4,3]=R[4]=0
4240 U[4,4]=-S3*F
4990 RETURN
5000 REM *** SUB-PROGRAM "ADAPT" ***
5010 U[1,5]=U[2,5]=U[3,5]=U[4,5]=0
5990 RETURN

```

NEWTON

```

3000 REM *** SUB-PROGRAM "PRINT" ***
3010 S1=8
3020 S3=5
3030 C0=0
3040 FOR I=1 TO N
3050 H=D[I+1]-D[I]
3060 C0=C0+.5*H*(V[I,1]+V[I+1,1]+.166667*H*
      *(V[I,2]-V[I+1,2]))
3070 NEXT I
3080 Q=S3*(S1*C0+V[N5,3])/(1-V[N5,4])
3090 PRINT "      INTEG=";C0;"Q9=";Q9;"Q=";Q
3100 PRINT " Y(0)=";V[1,1];"Y(1)=";V[N5,1];"W(1)=";V[N5,3]
3110 PRINT " Z(1)=";V[N5,4];"Q'=";Q[N2+2];"I9=";I9;"J9=";J9
3120 PRINT
3130 Q0=200
3140 ES=.0001
3150 GOSUB 5000
3160 IF P>0 THEN 3990
3170 PRINT "      X          Y(X)          V(X)          W(X)
3180 PRINT "      -----
3190 FOR I=1 TO N5
3200 PRINT USING 3210;D[I],V[I,1],V[I,2],V[I,3],V[I,4]
3210 IMAGE 8X,D,D,4(4X,D,5D)
3220 NEXT I
3230 STOP
3990 RETURN

```

ERROR

```

2000 REM *** SUB-PROGRAM "FUNC-G" ***
2010 G[1,6]=V[1,2]
2020 G[2,6]=V[1,3]
2030 G[3,6]=V[1,4]-1
2040 G[4,6]=V[5,2]
2990 RETURN
3000 REM *** SUB-PROGRAM "FUNC-F" ***
3010 READ S0,S1,S2,S3,S4,S5,S6
3020 DATA 10,8,5,5,-4,.5,.05
3030 RESTORE 3010
3040 Z0=1/(1+S6*(S4+Q[1]))
3050 E=EXP(-(S4+Q[1])*Z0)
3060 F=1/(S5+E)
3070 Q[1]=Q[2]
3080 Q[2]=S0*((S1+S2)*Q[1]-S2*Q[3]-Q9*F*Q[4])
3090 Q[3]=S2*(Q[1]-Q[3])
3100 Q[4]=-S3*F*Q[4]
3990 RETURN
4000 END

```

Пример 3

BPR-Q

```

1000 REM *** SUB-PROGRAM "INPUT" ***
1010 READ N,N0,M0,Q9,H9,I9,J9,H8,E0,E4
1020 DATA 10,4,2,0,.2,1,1,1,.0001,.0001
1030 READ E1,E2,E3,N8,N9,D2,I2,I7,N7,I3
1040 DATA .001,1.25,1,2,7,.2,1,2,3,0
1050 FOR I=1 TO N+1
1060 D[I]=(I-1)/N
1070 V[I,1]=V[I,2]=V[I,3]=V[I,4]=0
1080 H[I,1]=H[I,2]=H[I,3]=H[I,4]=.1
1090 NEXT I
1100 PRINT " TWO POINT BOUNDARY VALUE PROBLEM : "
1110 PRINT " 0<X<1, Y' = U "
1120 PRINT " U' = V "
1130 PRINT " V' = W "
1140 PRINT " W' = [V+Q/(1-Y)+2]/S0"
1150 PRINT " X=0: U=W=0; X=1: Y=U=0"
1160 PRINT " S0=.0001"
1170 PRINT
1990 RETURN
2000 END

```


MATRIC

```

3000 REM *** SUB-PROGRAM "BOND" ***
3010 G[1,1]=G[1,3]=G[1,4]=G[1,5]=0
3020 G[2,1]=G[2,2]=G[2,3]=G[2,5]=0
3030 G[3,2]=G[3,3]=G[3,4]=G[3,5]=0
3040 G[4,1]=G[4,3]=G[4,4]=G[4,5]=0
3050 G[1,2]=G[2,4]=G[3,1]=G[4,2]=1
3060 G[1,6]=V[1,2]
3070 G[2,6]=V[1,4]
3080 G[3,6]=V[N5,1]
3090 G[4,6]=V[N5,2]
3990 RETURN
4000 REM *** SUB-PROGRAM "JACOB" ***
4010 S0=.0001
4020 Z0=1/(1-Q[1])^2
4030 Q[1]=Q[2]
4040 U[1,1]=U[1,3]=U[1,4]=R[1]=0
4050 U[1,2]=U[2,3]=U[3,4]=1
4060 Q[2]=Q[3]
4070 U[2,1]=U[2,2]=U[2,4]=R[2]=0
4080 Q[3]=Q[4]
4090 U[3,1]=U[3,2]=U[3,3]=R[3]=0
4100 Q[4]=(Q[3]+Q9*Z0)/S0
4110 U[4,1]=2*Q9*Z0/(1-Q[1])/S0
4120 U[4,2]=U[4,4]=0
4130 U[4,3]=1/S0
4140 R[4]=Z0/S0
4990 RETURN
5000 REM *** SUB-PROGRAM "ADAPT" ***
5010 U[1,5]=U[2,5]=U[3,5]=U[4,5]=0
5990 RETURN
6000 END

```

NEWTON

```

3000 REM *** SUB-PROGRAM "PRINT" ***
3010 S0=.0001
3020 PRINT USING 3030;V[1,1],Q9,Q[N2+2]
3030 IMAGE 10X,"Y(0)=",D.5D,4X,"Q9=",D.5D,4X,"Q'=",SD.5DE
3040 PRINT USING 3050;Q9/S0,I9,J9
3050 IMAGE 10X,"S=",D.5DE,4X,"I9=",D.D,4X,"J9=",D.D
3060 PRINT
3070 E5=.0001
3080 GOSUB 4000
3090 IF P>0 THEN 3990
3100 PRINT "          X          Y(X)          U(X)
3110 PRINT "          -----
3120 FOR I=1 TO N5
3130 PRINT USING 3140;D[I],V[I,1],V[I,2],V[I,3],V[I,4]
3140 IMAGE 8X,D.5D,4X,D.5D,4X,SD.5D,2(4X,SD.5DE)
3150 NEXT I
3160 STOP
3990 RETURN

```

ERROR

```

2000 REM *** SUB-PROGRAM "FUNC-G" ***
2010 G[1,6]=V[1,2]
2020 G[2,6]=V[1,4]
2030 G[3,6]=V[N5,1]
2040 G[4,6]=V[N5,2]
2990 RETURN
3000 REM *** SUB-PROGRAM "FUNC-F" ***
3010 S0=.0001
3020 O[1]=Q[2]
3030 O[2]=Q[3]
3040 O[3]=Q[4]
3050 O[4]=(Q[3]+Q9/(1-Q[1])*2)/S0
3990 RETURN
4000 END

```

ПРИЛОЖЕНИЕ К § 9

ТЕКСТ ПРОГРАММЫ SOLV

SOLV

```

100 COM D[102],V[51,2],H[51,2]
110 COM N0,M0,N,Q9,H9,I9,J9,H8,E0,N8
120 COM N9,E1,E2,E3,I2,I7,N7,D2,E4,I3
130 COM I0,I1,I6,I8,J8,K9,N1,N2,N5,N6,L5
140 COM G[2,6],Y[102,2],P[102,2],H5,H6,H7
150 COM A[51,8],F[153,2],Q[9],Q7,P7,Q8,P8
160 DIM W[5,9],B[51,4]
1000 L0=2+(I9<N+2)
1010 L9=3*N0
1020 FOR K=1 TO N0
1030 FOR J=1 TO N0
1040 W[K,J]=G[K,J]
1050 W[K,J+N2]=G[K,J+N0]
1060 NEXT J
1070 W[K,L9+1]=G[K,N2+1]
1080 W[K,L9+2]=G[K,N2+2]
1090 W[K,L9+3]=0
1100 NEXT K
1110 IF L0=2 THEN 1520
1120 FOR I=1 TO N
1130 FOR K=1 TO N0
1140 F[I+N6,K]=0
1150 NEXT K
1160 NEXT I
1170 J0=J9+1
1180 IF I9>1 THEN 1260

```

SOLV

```

1190 FOR K=1 TO N0
1200 FOR J=J0 TO N0
1210 W[K,J-1]=W[K,J]
1220 NEXT J
1230 W[K,N0]=0
1240 W[K,L9+3]=G[K,J9]
1250 NEXT K
1260 IF I9=N5 THEN 1360
1270 FOR K=1 TO N0
1280 K1=N2*(K-1)
1290 F[N6+I9,K]=A[I9,K1+J9]
1300 FOR J=J0 TO N2
1310 A[I9,K1+J-1]=A[I9,K1+J]
1320 NEXT J
1330 A[I9,K1+N2]=0
1340 NEXT K
1350 IF I9=1 THEN 1520
1360 FOR K=1 TO N0
1370 K2=N2*(K-1)+N0
1380 F[I9-1+N6,K]=A[I9-1,K2+J9]
1390 FOR J=J0 TO N0
1400 A[I9-1,K2+J-1]=A[I9-1,K2+J]
1410 NEXT J
1420 A[I9-1,K2+N0]=0
1430 NEXT K
1440 IF I9<N5 THEN 1520
1450 FOR K=1 TO N0
1460 FOR J=J0 TO N0
1470 W[K,J+N2-1]=W[K,J+N2]
1480 NEXT J
1490 W[K,L9]=0
1500 W[K,L9+3]=G[K,J9+N0]
1510 NEXT K
1520 N3=L9+L0
1530 FOR I=1 TO N
1540 M8=N0+(I>I9)
1550 FOR K=1 TO M8
1560 FOR J=N1 TO N2
1570 W[K,J]=0
1580 NEXT J
1590 NEXT K
1600 FOR K=1 TO N0
1610 K1=N2*(K-1)
1620 FOR J=1 TO N2
1630 W[K+M8,J]=A[I,K1+J]
1640 NEXT J
1650 FOR J=1 TO N0
1660 W[K+M8,N2+J]=0
1670 NEXT J
1680 FOR M=1 TO L0
1690 W[K+M8,L9+M]=F[I+N5*(M-1),K]
1700 NEXT M
1710 NEXT K
1720 L1=N0-(I=I9)

```

SOLV

```
1730 N4=N0+M8
1740 IF I=N THEN 2010
1750 GOSUB 3050
1760 FOR K=1 TO L1
1770 K1=N2*(K-1)
1780 FOR J=1 TO N2
1790 A[I,K1+J]=W[K,J]
1800 NEXT J
1810 K2=N0*(K-1)
1820 FOR J=1 TO N0
1830 B[I,K2+J]=W[K,N2+J]
1840 NEXT J
1850 FOR M=1 TO L0
1860 F[I+N5*(M-1),K]=W[K,L9+M]
1870 NEXT M
1880 IF L0=3 THEN 1900
1890 F[I+N6,K]=F[I,K]
1900 NEXT K
1910 M9=N1-(I<I9)
1920 FOR K=1 TO M9
1930 FOR J=1 TO N0
1940 W[K,J]=W[L1+K,L1+J]
1950 W[K,J+N2]=W[L1+K,N2+J]
1960 NEXT J
1970 FOR M=1 TO L0
1980 W[K,L9+M]=W[L1+K,L9+M]
1990 NEXT M
2000 NEXT K
2010 NEXT I
2020 N3=N3-N0
2030 L1=N4
2040 FOR K=1 TO M8
2050 FOR J=N1 TO N3
2060 W[K,J]=W[K,J+N0]
2070 NEXT J
2080 NEXT K
2090 FOR K=1 TO N0
2100 FOR M=1 TO L0
2110 W[K+M8,N2+M]=W[K+M8,L9+M]
2120 NEXT M
2130 NEXT K
2140 IF I9<N OR L0=2 THEN 2310
2150 N3=N2+2
2160 IF I9=N5 THEN 2260
2170 FOR K=1 TO N0
2180 FOR J=N0 TO N3
2190 W[K,J]=W[K,J+1]
2200 NEXT J
2210 FOR M=1 TO 3
2220 W[K+N0,N2+M-1]=W[K+N0,N2+M]
2230 NEXT M
2240 NEXT K
```

SOLV

```

2250 GOTO 2310
2260 FOR K=1 TO N2
2270 FOR J=N2 TO N3
2280 W[K,J]=W[K,J+1]
2290 NEXT J
2300 NEXT K
2310 GOSUB 3050
2320 FOR M=1 TO K9
2330 L=N2+2-(M=1)*((I9=N5)+(I9=N))+ (M=2)*((I9<N)-(L0=2))
2340 W[N4,L]=W[N4,L]/W[N4,N4]
2350 K1=N4-1
2360 FOR K=K1 TO 1 STEP -1
2370 C0=W[K,L]
2380 K2=K+1
2390 FOR J=N4 TO K2 STEP -1
2400 C0=C0-W[K,J]*W[J,L]
2410 NEXT J
2420 W[K,L]=C0/W[K,K]
2430 NEXT K
2440 L4=N0-(I9=N)
2450 FOR K=1 TO L4
2460 F[N+N5*M,K]=W[K,L]
2470 NEXT K
2480 L5=N0-(I9=N5)
2490 FOR K=1 TO L5
2500 F[N5+N5*M,K]=W[K+L4,L]
2510 NEXT K
2520 Q[N2+M]=-(M=2)
2530 IF L0=2 THEN 2550
2540 Q[N2+M]=W[N4,L]
2550 K0=N-1
2560 FOR I=1 TO K0
2570 L1=N0-(I=I9)
2580 M7=I+N5*M
2590 FOR K=1 TO L1
2600 C0=F[M7,K]
2610 IF I9<N5 OR L0=2 THEN 2630
2620 B[I,N0*K]=F[I,K]
2630 K1=N0*(K-1)
2640 FOR J=1 TO N0
2650 C0=C0-B[I,K1+J]*W[L4+J,L]
2660 NEXT J
2670 IF I9>N THEN 2690
2680 C0=C0-F[I,K]*W[N4,L]
2690 F[M7,K]=C0
2700 NEXT K
2710 NEXT I
2720 FOR I=K0 TO 1 STEP -1
2730 L1=N0-(I=I9)
2740 L2=N2*(L1-1)+L1
2750 L3=N0-(I=I9-1)
2760 M7=I+N5*M
2770 FOR K=1 TO L1
2780 K1=N2*(K-1)+L1
2790 C0=F[M7,K]

```

S Q L V

```
2800 FOR J=1 TO L3
2810 C0=C0-A[I,K1+J]*F[M7+1,J]
2820 NEXT J
2830 F[M7,K]=C0
2840 NEXT K
2850 F[M7,L1]=F[M7,L1]/A[I,L2]
2860 K1=L1-1
2870 FOR K=K1 TO 1 STEP -1
2880 C0=F[M7,K]
2890 K2=K+1
2900 K3=N2*(K-1)
2910 FOR J=L1 TO K2 STEP -1
2920 C0=C0-A[I,K3+J]*F[M7,J]
2930 NEXT J
2940 F[M7,K]=C0/A[I,K3+K]
2950 NEXT K
2960 NEXT I
2970 IF L0=2 THEN 3030
2980 M7=I9+N5*M
2990 FOR K=N0 TO J0 STEP -1
3000 F[M7,K]=F[M7,K-1]
3010 NEXT K
3020 F[M7,J9]=-(M=2)
3030 NEXT M
3040 CHAIN "NEWTON"
3050 REM *** SUB-PROGRAM "GAUSS" ***
3060 FOR M=1 TO L1
3070 C0=0
3080 FOR K=M TO N4
3090 IF ABS(W[K,M])<ABS(C0) THEN 3120
3100 C0=W[K,M]
3110 K0=K
3120 NEXT K
3130 FOR J=M TO N3
3140 Q[J]=W[K0,J]
3150 W[K0,J]=W[M,J]
3160 W[M,J]=Q[J]
3170 NEXT J
3180 K1=M+1
3190 FOR K=K1 TO N4
3200 R=W[K,M]/C0
3210 FOR J=K1 TO N3
3220 W[K,J]=W[K,J]-R*Q[J]
3230 NEXT J
3240 NEXT K
3250 NEXT M
3260 RETURN
3270 END
```

Пример 1

BPR-Q

```

1000 REM *** SUB-PROGRAM "INPUT" ***
1010 READ N,N0,M0,Q9,H9,I9,J9,H8,E0,E4
1020 DATA 5,4,4,0,.2,6,1,1,.0001,.0001
1030 READ E1,E2,E3,N8,N9,D2,I2,I7,N7,I3
1040 DATA .001,1.25,1,2,7,.2,1,1,3,0
1050 FOR I=1 TO N+1
1060 D[I]=(I-1)/N
1070 V[I,1]=V[I,2]=V[I,3]=0
1080 V[I,4]=1
1090 H[I,1]=H[I,2]=H[I,3]=H[I,4]=.25
1100 NEXT I
1110 PRINT " TWO POINT BOUNDARY VALUE PROBLEM : "
1120 PRINT " 0<X<1 "
1130 PRINT " Y'=V"
1140 PRINT " V'=S0*[(S1+S2)*Y-S2*W-Q*F(Y+S4)*Z]"
1150 PRINT " W'=S2*(Y-W)"
1160 PRINT " Z'=-S3*F(Y+S4)*Z"
1170 PRINT " X=0 : V=W=0, Z=1"
1180 PRINT " X=0 : V=0"
1190 PRINT " F(U)=1/[S5+EXP(-U/(1+S6*U))]"
1200 PRINT
1210 PRINT " S0=10, S1=8, S2=5, S3=5, S4=-4, S5=.5, S6=.05"
1220 PRINT
1990 RETURN
2000 END

```

MATRIC

```

3000 REM *** SUB-PROGRAM "BOND" ***
3010 FOR I=1 TO 4
3020 FOR J=1 TO 9
3030 G[I,J]=0
3040 NEXT J
3050 NEXT I
3060 G[1,2]=G[2,3]=G[3,4]=G[4,6]=1
3070 G[1,10]=V[1,2]
3080 G[2,10]=V[1,3]
3090 G[3,10]=V[1,4]-1
3100 G[4,10]=V[N5,2]
3990 RETURN

```

Пример 2

BPR-Q

```

1000 REM *** SUB-PROGRAM "INPUT" ***
1010 READ N,N0,M0,Q9,H9,I9,J9,H8,E0,E4
1020 DATA 20,2,2,.2,.05,22,0,1,.0001,.0001
1030 READ E1,E2,E3,N8,N9,D2,I2,I7,N7,I3
1040 DATA .001,1,1,5,10,.2,0,1,4,0
1050 FOR I=1 TO N+1
1060 D(I)=(I-1)/N
1070 V(I,1)=V(I,2)=0
1080 H(I,1)=H(I,2)=.2
1090 NEXT I
1100 PRINT " TWO POINT BOUNDARY VALUE PROBLEM : "
1110 PRINT " 0<X<1 "
1120 PRINT " Y'=T*V, P0=6.28319, T=P0/S1 "
1130 PRINT " V'=T*(S9*SIN(P0*X)-S0+2*Y+Q*Y+2) "
1140 PRINT " Y(0)=Y(1), V(0)=V(1) "
1150 PRINT
1160 PRINT " S0=SQR(2), S1=1, S9=1 "
1170 PRINT
1990 RETURN
2000 END

```

MATRIC

```

3000 REM *** SUB-PROGRAM "BOND" ***
3010 G(1,1)=G(2,2)=1
3020 G(1,3)=G(2,4)=-1
3030 G(1,2)=G(1,4)=G(1,5)=0
3040 G(2,1)=G(2,3)=G(2,5)=0
3050 G(1,6)=V(1,1)-V(N5,1)
3060 G(2,6)=V(1,2)-V(N5,2)
3990 RETURN
4000 REM *** SUB-PROGRAM "JACOB" ***
4010 S0=SQR(2)
4020 S1=1
4030 S9=1
4040 S2=S0+2
4050 T=6.28319/S1
4060 U(1,1)=U(2,2)=R(1)=0
4070 U(1,2)=T
4080 O(1)=T*Q(2)
4090 U(2,1)=T*(2*Q9*Q(1)-S2)
4100 R(2)=T*Q(1)+2
4110 O(2)=T*(S9*SIN(6.28319*D)+Q9*Q(1)+2-S2*O(1))
4990 RETURN
5000 REM *** SUB-PROGRAM "ADAPT" ***
5010 U(1,3)=0
5020 U(2,3)=6.28319*T*S9*COS(6.28319*D)
5990 RETURN
6000 END

```


NEWTON

```
3000 REM *** SUB-PROGRAM "PRINT" ***
3010 S0=SQR(2)
3020 S1=1
3030 S9=1
3040 S2=S0+2
3050 P0=6.28319
3060 T=P0/S1
3070 J0=1
3080 GOSUB 4500
3090 PRINT " Q=";Q9;"AMPL=";C0;"X0=";X0
3100 PRINT " V0=";V0;"I9=";I9;"J9=";J9;"Q'=";Q[N2+2]
3110 PRINT
3120 A0=S9/(S0-S1)/(S0+S1)
3130 A1=1/(2*S1-S0)/(2*S1+S0)
3140 A2=(2/S0+2-A1)/(S0-S1)/(S0+S1)
3150 A3=A1/(S0-3*S1)/(S0+3*S1)
3160 DEF FNU(X)=A0*SIN(P0*X)
3170 DEF FNV(X)=.5*A0+2*(1/S0+2+A1*COS(2*P0*X))
3180 DEF FNW(X)=.5*A0+3*(A2*SIN(P0*X)+A3*SIN(3*P0*X))
3190 PRINT " X Y(X) U(X)"
3200 PRINT " -----"
3210 FOR I=1 TO N5
3220 D=D[I]
3230 Y=FNU(D)+Q9*(FNV(D)+Q9*FNW(D))
3240 PRINT USING "8X,D.DD,2(5X,SD.5D)";D[I],V[I,1],Y
3250 NEXT I
3260 STOP
3490 RETURN
3500 REM *** SUB-PROGRAM "JACOBO" ***
3510 Q[1]=T*Q[2]
3520 Q[2]=T*(S9*SIN(6.28319*D)+Q9*Q[1]+2-S2*Q[1])
3990 RETURN
```

NEWTON

```
4500 REM *** SUB-PROGRAM "AMPL" ***
4510 FOR I=1 TO N5
4520 FOR K=1 TO N0
4530 Q[K]=V[I,K]
4540 NEXT K
4550 D=D[I]
4560 GOSUB 3500
4570 FOR K=1 TO N0
4580 F[I,K]=Q[K]
4590 NEXT K
4600 NEXT I
4610 V0=0
4620 FOR I=1 TO N
4630 H=D[I+1]-D[I]
4640 V0=V0+.5*H*(V[I,J0]+V[I+1,J0]+.166667*H*
X(F[I,J0]-F[I+1,J0]))
4650 NEXT I
```

NEWTON

```

4660 V0=V0/(D[N5]-D[1])
4670 C0=ABS(V[1,J0]-V0)
4680 X0=D[1]
4690 FOR I=1 TO N
4700 IF ABS(V[I+1,J0]-V[I,J0])<C0 THEN 4730
4710 C0=ABS(V[I+1,J0]-V[I,J0])
4720 X0=D[I+1]
4730 IF SGN(F[I,J0])=SGN(F[I+1,J0]) THEN 4890
4740 H=D[I+1]-D[I]
4750 H0=H*F[I,J0]
4760 A=V[I,J0]-V[I+1,J0]+H0
4770 B=V[I+1,J0]-V[I,J0]-H*F[I+1,J0]
4780 C=2*A-B
4790 IF C>0 THEN 4820
4800 T=H0/(C-SQR(ABS(C^2+3*(B-A)*H0)))
4810 GOTO 4830
4820 T=H0/(C+SQR(ABS(C^2+3*(B-A)*H0)))
4830 IF T >= 0 AND T <= 1 THEN 4850
4840 T=H0/(3*(A-B)*T)
4850 C1=V[I,J0]*(1-T)+V[I+1,J0]*T-V0+T*(1-T)*
      X(A*(1-T)+B*T)
4860 IF ABS(C1)<C0 THEN 4890
4870 C0=ABS(C1)
4880 X0=D[I]+H*T
4890 NEXT I
4990 RETURN

```

ERROR

```

2000 REM *** SUB-PROGRAM "FUNC-G" ***
2010 G[1,6]=V[1,1]-V[N5,1]
2020 G[2,4]=V[1,2]-V[N5,2]
2990 RETURN
3000 REM *** SUB-PROGRAM "FUNC-F" ***
3010 S0=SQR(2)
3020 S1=1
3030 S9=1
3040 S2=S0^2
3050 T=6.28319/S1
3060 Q[1]=T*Q[2]
3070 Q[2]=T*(S9*SIN(6.28319*D)+Q9*Q[1]^2-S2*Q[1])
3990 RETURN
4000 END

```

СПИСОК ЛИТЕРАТУРЫ

1. Ортега Дж., Рейнболд В. Итерационные методы решения нелинейных систем уравнений со многими неизвестными.— М., 1975.
2. Оден Дж. Конечные элементы в нелинейной механике сплошных сред.— М.: Мир, 1976.
3. Kubicek M. Dependence of Solution of Nonlinear System on Parameter // ACM Trans. on Math. Software.— 1976.— V. 2, N 1.— P. 98—107.
4. Kubicek M., Marec M. Computational Methods in Bifurcation Theory and Dissipative Structures.— New York — Berlin — Tokyo: Springer-Verlag, 1983.
5. Keller H. B. Numerical Methods for Two-Point Boundary Value Problems.— Ontario — London, Waltham, Mass — Toronto: Blaisdell Publishing Co, 1968.
6. Беллман Р., Калаба Р. Квазилинеаризация и нелинейные краевые задачи.— М.: Мир, 1968.
7. Годунов С. К. О численном решении краевых задач для систем линейных обыкновенных дифференциальных уравнений // Успехи мат. наук.— 1961.— Т. 16, вып. 3.— С.171—174.

8. Бор К. Практическое руководство по сплайнам.— М.: Радио и связь, 1985.
9. Завьялов Ю. С., Квасов Б. И., Мирошниченко В. Л. Методы сплайн-функций.— М.: Наука, 1980.
10. Лукьянова Р. Г., Фадеев С. И., Шведова К. В. Расчет статистических параметров механической модели пленочного электростатического реле // Вычислительные системы.— Новосибирск, 1980.— Вып. 40.— С. 3—50.
11. Доронин В. П., Лукьянова Р. Г., Фадеев С. И. Численное построение всех решений краевой задачи для системы нелинейных дифференциальных уравнений // Вычислительные системы.— Новосибирск, 1977.— Вып. 72.— С. 99—114.
12. Фадеев С. И. Об электростатическом притяжении пленочной металлизированной ленты // Вычислительные системы.— Новосибирск, 1981.— Вып. 84.— С. 64—73.
13. Лукьянова Р. Г., Фадеев С. И., Шеплев В. С. К расчету экзотермической реакции первого порядка на зерне катализатора // Вычислительные системы.— Новосибирск, 1981.— Вып. 87.— С. 99—113.
14. Фадеев С. И. Расчет характеристик многослойной емкостной структуры // Вычислительные системы.— Новосибирск, 1982.— Вып. 95.— С. 66—78.
15. Фадеев С. И. О решении системы трансцендентных уравнений с параметром методом Ньютона.— Новосибирск, 1984.— 25 с.— (Препринт/СО АН СССР. Институт математики; № 72).
16. Фадеев С. И. О решении системы трансцендентных уравнений с параметром методом Ньютона // Вычислительные системы.— Новосибирск, 1985.— Вып. 108.— С. 78—93.
17. Фадеев С. И. О применении параметризации при решении нелинейных краевых задач // Вычислительные системы.— Новосибирск, 1987.— Вып. 121.— С. 102—124.
18. Зеленик Т. И. Качественная теория краевых задач для квазилинейных уравнений второго порядка параболического типа.— Новосибирск, 1972.
19. Luss D. Some further observation concerning multiplicity and stability of distributed parameter system // CES.— 1972.— V. 27.— P. 2299.
20. Емельянов И. Д., Шеплев В. С., Слинько М. Г. Число стационарных состояний экзотермических процессов в организованном псевдоожигенном слое катализатора // Докл. АН СССР.— 1985.— Т. 221, № 2.— С. 402—404.
21. Тихиро Х. Вынужденные колебания в нелинейных системах.— М.: Изд-во иностр. лит., 1957.
22. Инструкция по использованию системой разделения времени BASIC 2000 F.— Новосибирск: ВЦ СО АН СССР, 1975.