7259

СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА

Р5 - 7259

Л.Александров

ПРОГРАММА ДЛЯ РЕШЕНИЯ НЕЛИНЕЙНЫХ СИСТЕМ УРАВНЕНИЙ НА ОСНОВЕ РЕГУЛЯРИЗОВАННЫХ ИТЕРАЦИОННЫХ ПРОЦЕССОВ ГАУССА-НЬЮТОНА

1973

ЛАБОРАТОРИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ И АВТОМАТИЗАЦИИ

P5 - 7259

Л.Александров

ПРОГРАММА ДЛЯ РЕШЕНИЯ НЕЛИНЕЙНЫХ СИСТЕМ УРАВНЕНИЙ НА ОСНОВЕ РЕГУЛЯРИЗОВАННЫХ ИТЕРАЦИОННЫХ ПРОЦЕССОВ ГАУССА-НЬЮТОНА

СОДЕРЖАНИЕ

Различные вопросы автоматизации физических исследований приводят к разработке программ для решения нелинейных систем уравнений, обладающих расширенными алгоритмическими возможностями.

В настоящей работе предлагается составной алгоритм для решения нелинейных систем уравнений, основанный на α-μ-регуляризованных итерационных процессах Гаусса-Ньютона /I/. В этом алгоритме охвачен ряд важных для вычислительной практики на ЭВМ R-процессов, таких как авторегуляризованный процесс R_{ε_n} /2/, процесс с экспоненциально убывающими операторами регуляризации /3,4/, процесс с компенсированной регуляризацией /I/ и процесс с "наилучшей поправкой" (который описывается в п. 4 настоящей работы).

Предлагаемый в работе алгоритм реализован в написанной на ФОРТРАНе программе (SUBRØUTINE REGN), которую можно непосредственно применять, в частности, на ЭВМ БЭСМ-6 и машинах CDC серии 6000. Тело программы REGN приводится в приложении I.

Кроме возможности использования различных R-процессов типа Гаусса-Ньютона, в алгоритме программы REGN предусмотрены относительно широкие возможности в выборе способа построения матриц Якоби, способа прерывания итерационных процессов; в выборе типа решаемой нелинейной системы уравнений, а также в выборе норм для векторов и матриц.

В основном программа REGN предназначена для построения устойчивых итерационных процессов относительно искажений входных данных и ошибок округлений при работе ЭВМ. К необходимости построения таких процессов приводит, в частности, решение нелинейных систем, имеющих непростые нули.

Расширенные возможности программы REGN можно использовать при создании специальных алгоритмов для решения "потоков нелинейных задач", получаемых при обработке результатов физического эксперимента[4,5].

Программу REGN можно также использовать при автоматизации исследования "непрерывных задач"дифференциальные, интегральные уравнения, краевые задачи и др.) на основе системы человек - машина[6].

П. I. Основные модификации алгоритма программы REGN

Пусть R^N - N-мерное вещественное пространство точек $(x_1, \ldots, x_N)$. Пусть в некоторой выпуклой области $D_f \subseteq R^N$ заданы $\bar{M}$ нелинейных вещественных функций $f_j(x_1, \ldots, x_N), j=1, \ldots, \bar{M}$, обладающих частными производными по всем компонентам $x_1, \ldots, x_N$.

В области D_f рассматриваем нелинейную систему уравнений

$$\left.\begin{array}{c} f_1(x_1, \ldots, x_N) = y_1 \\ \vdots \\ f_{\bar{M}}(x_1, \ldots, x_N) = y_{\bar{M}} \end{array}\right\} \qquad (I)$$

относительно неизвестных $x_1, \ldots, x_N$, где $y_1, \ldots, y_{\bar{M}}$ - заданные вещественные числа.

Система (I) в программе REGN решается на основе производимой конечной части некоторого регуляризованного итерационного процесса типа Гаусса-Ньютона[I] (конечная часть -

это последовательность приближений, начинающаяся с заданного начального приближения x_0 и кончающаяся приближением x_{n^*} с достаточно большим номером итерации n^*).

R -процессы Гаусса-Ньютона в программе REGN возможно выполнить в различных вариантах (модификациях). Алгоритм программы REGN удобно рассматривать как составленный из нескольких видов модификаций выполняемых итерационных процессов.

Основной вид модификаций (A -модификации алгоритма) представляют различные допустимые случаи R -процессов Гаусса-Ньютона.

Все способы дифференцирования, имеющие место при образовании матриц Якоби, составляют D -модификации алгоритма.

Различные способы прерывания итерационных процессов из класса A образуют G -модификации алгоритма.

Как S -модификации алгоритма понимаются все допустимые типы решаемых нелинейных систем (I).

И, наконец, все способы получения информации о найденном решении системы (I) понимаются как P -модификации алгоритма.

Кроме описанных основных модификаций в REGN осуществлен и ряд второстепенных, которые перечисляются в последующем подробном описании алгоритма.

Формальные параметры NP, M , N , K , KK , LT, KP, T , AD , S , TT , D , E , AM , AN программы REGN (см. приложение I), а также элементы общих (помеченных) блоков CINT, DROPX, WEPS ,WENORM, KSEL , LINT , NUITT и NL (см. приложение I) называются "параметрами управления программы REGN " (в дальнейшем обозначаются как ПУ).

Реализация той или иной модификации алгоритма осуществляется при помощи некоторого фактического набора из перечисленных ПУ. Будем различать два вида ПУ: "основные параметры управления" (ОПУ) – ПУ, которые являются формальными параметрами программы REGN , и "внутренние параметры управления" (ВПУ) – ПУ, являющиеся элементами упомянутых общих блоков.

Для упрощения при задании модификаций программы REGN предусмотрено "самоуправление" ВПУ, т.е. автоматическое присваивание параметрам ВПУ стандартных значений, отвечающих самым применяемым модификациям алгоритма.

Индикацией для включения режима самоуправления служит нулевое значение параметров ВПУ. Самоуправление дает возможность программисту не определять значения параметров ВПУ. В таком случае ВПУ имеют автоматически нулевые значения/7/ (для достижения этого эффекта при работе с мониторной системой SCØPE для ЭВМ CDC -6000 среди управляющих карт надо поставить карту SETCØRE/8/).

Программа REGN использует 8 вспомогательных подпрограмм, среди которых важное значение для ее алгоритма имеют TRAM , SYMIN и ØUTPUT . Одна из вспомогательных подпрограмм, а именно RELADI , составляется пользователем в зависимости от решаемой конкретной задачи. В программе RELADI вводится основная информация о решаемой задаче – уравнения системы (I) и частные производные $\frac{\partial f_j}{\partial x_i}$, если для них существуют аналитические выражения.

Программа REGN составлена из 6 крупных блоков-частей (обозначенных буквами от A до F , см. приложение I). В подготовительном блоке *A* ПУ программы переводятся на внутренний язык, а также реализуется режим самоуправления.

В блоке *B* производится первая итерация для некоторых из главных модификаций класса A .

Блок *C* осуществляет связывающие функции между блоками *B* и *D*.

В главном блоке *D* программы REGN выполняется основной итерационный цикл для всех модификаций класса A .

В блоке *E* реализованы способы прерывания итерационного процесса.

Блок *F* функционирует вместе с подпрограммой SYMIN при осуществлении обобщенного обращения симметричных матриц (см. П.6).

В следующих пунктах работы приводится подробное описание алгоритма программы REGN .

П. 2. Общая итерационная схема

Пусть $R^{\bar{M}}$ – $\bar{M}$-мерное вещественное пространство точек $(y_1, \dots, y_{\bar{M}})$.

Для описания алгоритма программы REGN удобнее использовать векторную запись системы (I):

$$fx = y \quad , \tag{2}$$

где

$x=(x_1,\dots,x_N)^T \in R^N$, $fx=(f_1(x),\dots,f_{\bar{M}}(x))^T \in R^{\bar{M}}$ и $y=(y_1,\dots,y_{\bar{M}})^T \in R^{\bar{M}}$

В дальнейшем будем предполагать, что в пространстве введена равномерная норма векторов с весами $\bar{g}_i > 0$ $(i=1,\dots,N)$

$$\|x\| = \max_i \bar{g}_i |x_i| \tag{3}$$

и подчиненная ей норма для квадратной матрицы $Z=[z_{ij}]$

$$\|Z\| = \max_i \sum_{j=1}^{N} \frac{1}{\bar{g}_i}|z_{ij}| \quad . \tag{4}$$

Нормы (3) и (4) являются основными в программе REGN . Веса $\bar{g}_i$ $(i=1,\dots,N)$ выведены как ВПУ и задаются при помощи общего блока WENФRM. В режиме самоуправления в REGN им присваиваются единицы.

Пусть $\mathcal{J}(x)=\left[\frac{\partial f_j}{\partial x_i}\right]_{\substack{i=1,\dots,N \\ j=1,\dots,\bar{M}}}$ обозначает матрицу Якоби для оператора f . Матрицу Якоби в точке x_n будем обозначать сокращенно с $\mathcal{J}_n$.

Все вошедшие в алгоритм программы REGN а-м- регуляризованные процессы Гаусса-Ньютона для решения уравнения (2) можно записать в общую итерационную схему

$$R:\ x_0\ ,\ x_{n+1} = x_n - \\ - v_n(\mathcal{J}_n^T G \mathcal{J}_n + \varepsilon_n U)^{-1}(E-\delta\varepsilon_n v_n(\mathcal{J}_n^T G\mathcal{J}_n+\varepsilon_n U)^{-1})\mathcal{J}_n^T G(f x_n - y), \tag{5}$$

$x_n \in D_f$, ($n=0,1,\dots,n^*$ - номер итерации, на которой процесс R прерывается).

Примененные в схеме (5) обозначения имеют следующий смысл:

$$v_n = \begin{bmatrix} 1-v_{1,n} & & 0 \\ & \ddots & \\ 0 & & 1-v_{N,n} \end{bmatrix}, \text{ где } v_{k,n}\in[0,1], (k=1,\dots,N)$$

- оператор мультипликативной регуляризации/I/;

$$\varepsilon_n U = \varepsilon_n \begin{bmatrix} u_1 & & 0 \\ & \ddots & \\ 0 & & u_N \end{bmatrix}, \text{ где } \varepsilon_n \in[0,\infty),\ t_k\in(0,\infty),$$

$(k=1,\dots,N)$ - оператор аддитивной регуляризации/I/. Числа ε_n имеют структуру

$$\varepsilon_n = \bar{\varepsilon}_n + \varepsilon_L \quad , \quad \bar{\varepsilon}_n, \varepsilon_L \in [0,\infty), \tag{6}$$

где $\bar{\varepsilon}_n$ - переменная по номеру итерации, а ε_L - постоянная;

$$G = \begin{bmatrix} g_1 & & 0 \\ & \ddots & \\ 0 & & g_{\bar{M}} \end{bmatrix}, \text{ где } g_k\in(0,\infty), (k=1,\dots,\bar{M})$$

- "матрица взвешивания" системы/I/. В обыкновенных случаях $G=E$ (случаи, когда $G\neq E$ описываются в П.9);

δ - число, которое принимает значения 0 или I. В первом случае (δ=0) из итерационной схемы (5) получается обыкновенный R-процесс Гаусса-Ньютона (обозначаемый в дальнейшем R^{GN}). Во втором случае (δ=I) из схемы (5) получается " R-процесс Гаусса-Ньютона с компенсированной регуляризацией"/I/ (обозначаемый в дальнейшем R_C^{GN}).

Общая схема (5) реализована в блоках *B* и *D* программы REGN . Эти блоки обращаются многократно к подпрограмме TRAM , где производится формирование матриц $\mathcal{J}_n^T G\mathcal{J}_n$ и $\mathcal{J}_n^T G(fx_n-y)$.

Матрицы v_n и U принимаются в программе REGN как ВПУ и задаются при помощи общих блоков DRФPX и WEPS соответственно. В режиме самоуправления элементам блока WEPS в REGN присваиваются единицы.

Аддитивная регуляризация в процессах (5) осуществляется на основе чисел $\mathcal{E}_n$ $(n=0,1,\dots,n^*)$, которые в основном определяются внутри программы REGN.

Постоянная $\mathcal{E}_L$ формулы (6) играет роль допустимого нижнего предела регуляризации (во многих случаях она может быть равна нулю). $\mathcal{E}_L$ принята как ВПУ и задается при помощи общего блока CINT : $\mathcal{E}_L = C(10)$ (на внутреннем языке REGN она обозначена как EPSLOW).

Для организации комбинированных a-u-регуляризованных процессов (5)/I/ выделена отдельная подпрограмма REML. Привлекая какой-нибудь из методов мультипликативной регуляризации (см., например,/9,10/) пользователь может составить свою программу REML для задания необходимых на каждой итерации специальных операторов v_n. В нашем варианте программы REML запрограммирована связь

$$v_{i,n} = a e^{-dn} + f, \quad (i=1,\dots,N).$$

Постоянные a, d и f выведены в общем блоке RM и при необходимости могут быть использованы в программе USER (см.п.11) как параметры типа ВПУ.

В программе REGN на каждой итерации и по схеме (5) вычисляются значения ряда критериев относительно процесса и находимого решения системы (I). Эти критерии разделяются на две группы. Первая группа - это "критерии цели". В нее входят функционалы

$$R\Phi \; (=\rho_n) = \| \mathcal{J}_n^T G (f x_n - y) \|,$$

$$\text{MAX DEFECT} = \| f x_n - y \|,$$

$$HI\ SQ = \sum_{i=1}^{\bar{M}} g_i (f_i x_n - y_i)^2,$$

(На внутреннем языке программы REGN они обозначены соответственно RΦM, RM и REV).

Вторая группа - это "критерии поведения процесса R ". В нее входят функционалы

$$TAU \; (= \tau_n) = \| \mathcal{J}_n^T G \mathcal{J}_n \|,$$

$$C\Phi ND = \| \mathcal{J}_n^T G \mathcal{J}_n + \mathcal{E}_n U \| \, \| (\mathcal{J}_n^T G \mathcal{J}_n + \mathcal{E}_n U)^{-1} \|,$$

$$EPS = \bar{\mathcal{E}}_n .$$

Критерии цели служат для интерпретации полученного решения. На их основе можно ответить на вопрос, каким является найденное решение программы REGN (ложным или настоящим/I/).

Критерии поведения дают информацию о степени вырожденности матрицы Якоби на каждой итерации и тем самым проливают свет на свойство устойчивости реализуемого процесса/II/. В этом отношении глубочайшей характеристикой является число CΦND /I2/.

П. 3. Авторегуляризованные процессы и процессы с экспоненциально убывающими операторами регуляризации

В программе REGN развиты только аддитивные регуляризации итерационных процессов типа (5). Они осуществляются различным способом задания чисел $\bar{\mathcal{E}}_n$ в выражении (6). В настоящем и следующем пункте описываются все модификации процессов R^{GN} и R_C^{GN}, входящие в класс A алгоритма.

A_I). Авторегуляризованный процесс R^{GN} первого типа(ARP-F).

Процесс ARP-F получается из общей схемы (5) при $\delta = 0$ и при $\bar{\mathcal{E}}_0 = \mathcal{E}_0$ ($\mathcal{E}_0$ - задаваемая постоянная),

$$\bar{\varepsilon}_n = \frac{\alpha_2}{2}\left(\sqrt{\tau_n^2 + 4N_o\rho_n} - \tau_n\right), \quad [2,1] \qquad (7)$$

$(n = 1, 2, \ldots, n^*)$,

где $N_o = \frac{\alpha_1}{\rho_o}(\varepsilon_o^2 + \varepsilon_o\tau_o)$ [2] , а $\alpha_1 \in [0,\infty)$ и $\alpha_2 \in (0,1]$ – постоянные для процесса.

Первая итерация процесса ARP-F проводится в блоке *B* программы REGN , остальные итерации – в блоке *D* .

Постоянные ε_o, α_1 и α_2 являются ОПУ и введены как формальные параметры E , AM , AN . На внутреннем языке $\bar{\varepsilon}_n$, τ_n , ρ_n , N_o , α_1 и α_2 обозначены,соответственно, EPS , TAU , RΦM , ENØ , ALPHA1 и ALPHA2 .

A$_2$) Авторегуляризованный процесс R^{GN} второго типа (ARP).

Процесс ARP отличается от процесса ARP-F только тем, что числа регуляризации $\bar{\varepsilon}_n$ задаются по авторегуляризационной связи (7) для всех $n = 0, 1, \ldots, n^*$. Постоянная N_o вычисляется, как и в случае процесса ARP-F , на основе задаваемой постоянной ε_o (управление E).

Модификация ARP осуществлена в блоке *D* программы REGN .

Относительно постоянной ε_o в процессах ARP-F и ARP предусмотрена также возможность для автоматического задания по формуле

$$\varepsilon_o = C \cdot \tau_o , \qquad (8)$$

которая реализована в блоке *A4* .

Постоянная C (на внутреннем языке обозначаемая EPSM ; в режиме самоуправления имеет значение 0.I) входит в програм-

му как ВПУ и задается при помощи $C(3)$ из общего блока CINT .

З а м е ч а н и е.

Постоянные α_1 и α_2 в формуле (7) используются для ослабления или усиления авторегуляризации. Обычно $\alpha_1 = \alpha_2 = 1$.

A$_3$) Процессы R^{GN} и R^{GN}_C с экспоненциально убывающими операторами регуляризации (ERP)

Процессы ERP получаются из общей схемы (5) при

$$\bar{\varepsilon}_n = |\alpha_1|\, e^{\alpha_2 n} \quad (n = 0, 1, \ldots, n^*) \quad [3,4] \qquad (9)$$

где α_1 и α_2 – постоянные, для которых $\alpha_1 \in (-\infty,\infty)$, $\alpha_2 \in (-\infty, 0]$. α_1 и α_2 задаются при помощи параметров ОПУ AM и AN ($AM = \alpha_1$, $AN = \alpha_2$).

Модификация ERP реализована в блоке *D* программы REGN .

Управление процессов ARP-F , ARP и ERP осуществляется при помощи ОПУ IT , E , AM и AN в соответствии со следующей таблицей.

Таблица I

МОД. \ ОПУ	IT	E	AM	AN
ARP-F	<0	≥ 0	≥ 0	>0
ARP	≥ 0	≥ 0	≥ 0	>0
ARP-F автом.выбор ε_o	<0	-1	≥ 0	>0
ARP автом.выбор ε_o	≥ 0	-1	≥ 0	>0
ERP для R^{GN}	≥ 0	0	≥ 0	≤ 0
ERP для R^{GN}_C	≥ 0	0	<0	≤ 0

П. 4. Процессы Гаусса-Ньютона с наилучшей поправкой (процессы RPBC)

Пусть задано некоторое множество положительных чисел

$$Q_S = \{ 0 \le \beta_0 < \beta_1 < \dots < \beta_S < \infty \} .$$

Процессы RPBC получаются из общей итерационной схемы (5), когда в качестве $\bar{\mathcal{E}}_n$ $(n=0,1,\dots,n^*)$ в формуле (6) используются числа $\mathcal{E}_n^{Q_S}$, определяемые из соотношения

$$\| f x_n(\mathcal{E}^{Q_S}) - y \|_e = \min_{\beta \in Q_S} \| f x_n(\beta) - y \|_e , \qquad (10)$$

где $x_n(\beta) = x_n -$

$$- v_n (J_n^T G J_n + \beta U)^{-1} (E - \delta \beta v_n (J_n^T G J_n + \beta U)^{-1}) J_n^T G (f x_n - y) .$$

Здесь использована сферическая норма (обозначенная на внутреннем языке как REV).

Реализованные в програме REGN процессы типа RPBC (понимающиеся как модификация A4) из класса A) основываются на специальном алгоритме для одновременного построения множества Q_S и проведения минимизации (10).

В своей работе этот алгоритм использует параметры ОПУ AD , S , TT и параметры типа ВПУ C(4) , C(14) из общего блока CINT (на внутреннем языке программы обозначены TADD и EBCL соответственно), а также LINT из общего блока LINT.

При построении множества Q_S используется рекуррентная зависимость

$$\beta_{K+1} = \beta_K + AD, \ (K = 0, 1, \dots, LINT), \text{ где} \qquad (11)$$
$$\beta_0 = -AD + EBCL .$$

На каждом шаге к этому внутреннему процессу вычисляется значение $\| f x_n(\beta_K) - y \|_e$. Выход из цикла (II) осуществляется в двух случаях:

I) достигнуто неравенство

$$\frac{| x_{i,n}(\beta_{K+1}) - x_{i,n}(\beta_K) |}{| x_{i,n}(\beta_K) |} \, 100 \le TT . \qquad (12)$$

(для всех компонент $x_{i,n}$ $(i = 1, \dots, N)$ вектора $x_n(\beta)$),
или

2) имеет место $K = LINT$.

Если этот критерий не выполнен, то внутренний процесс (II) повторяется около найденного грубого минимума с новым шагом AD , который вычисляется по формуле

$$(AD)_{\text{новый}} = S \times (AD)_{\text{старый}} . \qquad (13)$$

Изменение шага в (II) на основе (I3) повторяется до тех пор пока не сработает критерий выхода из цикла (II).

В тех случаях, когда заданный множитель S оказывается недостаточно малым и выработанный в (I3) новый шаг AD приводит к $\beta < 0$, начальное число β_0 в (II) строится по другой формуле:

$$\beta_0 = -AD + AD/TADD .$$

При этом программа REGN печатает индикацию "CORRECTED ITERATION"

В режиме самоуправления

EBCL = 0.0001 , TADD = 10 , LINT = 100 .

Отметим, что функция $\| f x_n(\beta) - y \|_e$, $\beta \in [0, \infty)$ в общем случае имеет более, чем один локальный минимум.

При работе с REGN предпочитается минимум, отвечающий самому маленькому β. В связи с этим необходимо проявлять некоторую осторожность при выборе параметров AD и S, а также проводить численные эксперименты для нахождения тех ПУ, которые обеспечивают хорошую работу процесса RPBC.

В программе REGN предусмотрено также автоматическое задание управления AD по формуле (8).

Для выигрыша в быстродействии при осуществлении процессов RPBC массив $ZL(N,N)$ используется как рабочий. В нем хранится матрица $\mathcal{J}_n^T G \mathcal{J}_n$, остающаяся постоянной в процессах минимизации (10).

Процессы RPBC осуществляются на основе ОПУ NN, AD, S, TT, E и AM в соответствии со следующей таблицей.

Таблица 2

RPBC \ ОПУ	NN	AD	S	TT	E	AM
RPBC для R^{GN}	N	>0	>0	>0	0	≥0
RPBC для R_c^{GN}	N	>0	>0	>0	0	<0
RPBC для R^{GN} автом. выбор AD	N	0	>0	>0	−1	≥0
RPBC для R_c^{GN} автом. выбор AD	N	0	>0	>0	−1	<0

П. 5. Образование матриц $\mathcal{J}_n^T G \mathcal{J}_n$ и $\mathcal{J}_n^T G(fx_n - y)$

Задание G и y в $\mathcal{J}_n^T G \mathcal{J}_n$ и $\mathcal{J}_n^T G(fx_n - y)$ связано с типизацией нелинейных систем. Информация о векторе fx_n содержится в составленной пользователем подпрограмме RELADI. Для образования матрицы Якоби $\mathcal{J}(x_n)$ также используется информация из подпрограммы RELADI.

Все функции для образования составных элементов матриц $\mathcal{J}_n^T G \mathcal{J}_n$ и $\mathcal{J}_n^T G(fx_n - y)$, как и сами эти матрицы выполняются в подпрограмме TRAM. Кроме описанных функций, подпрограмма TRAM выполняет две дополнительные: печатает, при необходимости, дефекты после восстановления на каждой итерации и вычисляет критерий цели REV. Для образования $\mathcal{J}_n$ в подпрограмме TRAM кроме дифференцирования по заданным формулам реализованы несколько способов численного дифференцирования. Способы дифференцирования составляют класс D - модификаций общего алгоритма программы REGN.

D_1) Дифференцирование по формулам.

Частные производные $\frac{\partial f_j}{\partial x_i}$ вычисляются по заданным в подпрограмме RELADI аналитическим выражениям.

D_2) Численное дифференцирование с равным шагом.

Частные производные вычисляются по формуле

$$\frac{\partial f_j}{\partial x_i} \simeq \frac{f_j[x_i + h_i] - f_j(x)}{h_i}, \qquad (14)$$

где $h_i > 0$ не зависимо от номера i. В (14) $f_j[x_i+h_i]$ обозначает $f_j(x_1, \dots, x_i + h_i, \dots, x_N)$;

Шаг h в формуле (I4) задается при помощи ОПУ D с равенством

$$h = |D| . \qquad (15)$$

D_3) Численное дифференцирование с шагом, зависящим от неизвестных.

Частные производные вычисляются по формуле (I4) с шагом дифференцирования, имеющим вид

$$h_i = c\,|x_i| , \qquad (16)$$

где $c > 0$ - постоянная для всего процесса, которая задается при помощи ОПУ D равенством

$$c = |D| - 100 . \qquad (17)$$

(Так, например, чтобы иметь $c = 0.001$, надо поставить D = =100.001). На внутреннем языке REGN и TRAM постоянная c обозначена как CN6.

D_4) Сглаженное численное дифференцирование с равным шагом.

Частные производные вычисляются по формуле /13, стр.600/

$$\frac{\partial f_j(x)}{\partial x_i} \approx \bigl(3 f_j[x_i+h_i] + 10 f_j(x) - 18 f_j[x_i-h_i] + 6 f_j[x_i-2h_i] - f_j[x_i-3h_i]\bigr)/(12 h_i) . \qquad (18)$$

Шаг дифференцирования h_i, как и в модификации D_2), не зависит от номера i. Он задается при помощи ОПУ D согласно равенству (I5).

D_5) Сглаженное численное дифференцирование с шагом, зависящим от неизвестных.

Частные производные в этом случае вычисляются по формуле (I8), но с шагом, задаваемым согласно (I6). Константа задается при помощи равенства (I7).

Управление перечисленными модификациями алгоритма программы REGN производится при помощи одного параметра D в соответствии со следующей таблицей.

Таблица 3

ОПУ \ МОД	D1)	D2)	D3)	D4)	D5)
D	0	> 0	< 0	> 100	<-100

П. 6. Обращение матрицы $J_n^T G J_n + \varepsilon_n U$

Обращение матрицы $S_n = J_n^T G J_n + \varepsilon_n U$ занимает центральное место в алгоритме программы REGN. Обращение выполняется при помощи специальной подпрограммы SYMIN (переработанный вариант библиотечной программы SYMINV /14/). Для обращения симметричной матрицы программа SYMINV использует метод Гаусса-Жордана с нормализацией/15, алгоритм 150/. Если матрица S_n является вырожденной (или достаточно близка к такому состоянию), ее обращение при помощи SYMIN невозможно и в программе SYMIN вырабатывается признак IFAIL=1. В этом случае в программе REGN производится автоматическое улучшение обусловленности S_n путем увеличения числа ε_n

до тех пор, пока ее обращение при помощи SYMIN окажется возможным (блоки *B* и *D* - BASIS SEGMENT).

Увеличение числа $\mathcal{E}_n$ в REGN производится при помощи рекурсивной связи, которая на внутреннем языке программы имеет вид

$$(EPS)_{новое} = CEPS \times ((EPS)_{старое} + EFALI). \tag{19}$$

Константы CEPS и EFALI являются ВПУ. Их можно менять, используя общий блок CINT (CEPS=C(1), EFALI=C(6)). В режиме самоуправления CEPS=5, EFALI=0.0001. Во всех случаях, когда применяется связь (19) и печать по итерации разрешена, программа REGN печатает индикацию „CORRECTED ITERATION".

Кроме описанного "обыкновенного обращения" матрицы S_n, алгоритм программы REGN позволяет проводить „обобщенное обращение" для S_n. Введем некоторые вспомогательные понятия.

Пусть задана симметрическая матрица S с размерностью q и пусть на ней отмечены k $(0 \le k \le q)$ "крестов" (под крестом понимаем вместе взятые строку и столбец, пересекающиеся на диагонали). Под "сжатием матрицы S на k крестов" понимается получение новой матрицы (обозначаемой как $\not{k}S$) путем удаления из матрицы S отмеченных крестов.

Очевидно, размерность матрицы $\not{k}S$ равна $q-k$.

Под "растяжением матрицы $(\not{k}S)$ на k нулевых крестов" понимается получение новой матрицы (обозначаемой в дальнейшем как $\uparrow_k(\not{k}S)$) путем прибавления на место удаленных при операции $\not{k}$ крестов новых крестов, состоящих из нулей. Размерностью матрицы $\uparrow_k(\not{k}S)$ снова является число q.

Обобщенное обращение в программе REGN состоит в построении матрицы

$$S_n^{(k)} = \uparrow_k((\not{k}S_n)^{-1}) \tag{20}$$

для заданных k крестов.

В тех случаях, когда $\mathcal{E}_n = 0$ и заданные для получения матрицы $S_n^{(k)}$ кресты вырождают матрицу $\mathcal{J}_n^T G \mathcal{J}_n$, описанное обобщенное обращение матрицы S_n является "g-обращением" /16/.

Действительно, $S_n^{(k)}$ в этом случае удовлетворяет соотношению $S_n S_n^{(k)} S_n = S_n$.

Обращение (20) имеет существенное применение при нахождении статистических ошибок в случае непростого решения (см. П.9).

При построении оператора $S_n^{(k)}$ используются блоки *CRINGE* и *ENLARGE* программы REGN, а также подпрограмма SYMIN.

Обращение (20) осуществляется на основе внешней информации об удаляемых крестах, которая задается при помощи элементов DX общего блока DROPX (ВПУ). Элементам DX, соответствующим удаляемым крестам, программой USER (см. П. 13, пример 3) присваивается значение 1. На основе отмеченных таким образом крестов в программе REGN производится обращение типа (20) для всех $S_n (n=0,1,\dots,n^*)$.

Обращение (20) отсутствует в общей итерационной схеме (5) так как в настоящее время оно производится при помощи внешней информации, зависящей от пользователя.

П. 7. Прерывания итерационного процесса

В блоке *E* программы REGN реализован ряд способов прерывания итерационного процесса (5), составляющих модификации класса G . Эти модификации алгоритма управляются при помощи параметров типа ОПУ IT , T и параметров типа ВПУ KGS и L из общего блока KSEL .

При работе с программой REGN можно применять следующие прерывания итерационного процесса.

G_I) Выход по достижении заданного числа итераций n^* .

Процесс (5) прерывается на итерации с номером

$$n^* = |IT| .$$

G_2) Выход по достижении заданного критерия цели или заданного числа итераций.

Процесс (5) прерывается на первой итерации (из заданных допустимых итераций), для которой выполняется неравенство

$$C \leq |T| ,$$

где C - определенный вид критерия цели (RФ , MAX DEFECT или HI SQ).

Число допустимых итераций задается как в случае G_I).

G_3) Выход по достижении заданного толеранса для решения или по заданному числу итераций.

Процесс (5) прерывается на первой из заданных n^* допустимых итераций, для которой неравенство

$$\frac{|x_{i,n+1} - x_{i,n}|}{|x_{i,n}|} 100 \leq |T|$$

выполнено одновременно для всех компонент $x_i (i=1,\dots,N)$.

Число допустимых итераций задается как в случае G_I).

G_4) Выход по нарушению монотонного убывания критерия цели или по заданному числу итераций.

Процесс (5) прерывается на первой итерации $\bar{n}$ (среди заданных n^* итераций), для которой выполняется неравенство

$$(C)_{\bar{n}} \leq (C)_{\bar{n}+1} ,$$

где C - определенный вид критерия цели (RФ , MAX DEF. или HI SQ).

Число итераций n^* задается как в случае G_I).

Имеют место еще два конъюнктивных типа выходов из программы REGN.

G_5) : G4) или G2) ,

G_6) : G4) или G3) .

Управлений модификаций класса G осуществляется в соответствии со следующей таблицей.

Таблица 4

ПУ \ МОД	G1)	G2)	G3)	G4)	G5)	G6)
L = 1		C = RФ		C = RФ	C = RФ	C = RФ
L = 2		C = M.D.		C = M.D.	C = M.D.	C = M.D.
L = 3		C = HI SQ		C = HI SQ	C = HISQ	C = HISQ
T	0	< 0	> 0	0	< 0	> 0
KGS	0	0	0	1	1	1

П. 8. Типизация решаемых нелинейных систем уравнений

Программа REGN применяется к решению разных типов нелинейных систем уравнений, что приводит к различию в структуре входных данных и к различию в построении нестандартной подпрограммы RELADI.

Входные данные задачи составляют вектор y, матрица G и параметры, от которых зависят связи f_j. Входные данные задаются при помощи массива YR.

Различные допустимые типы системы (I) образуют класс S -модификаций алгоритма программы REGN. Модификации S управляются при помощи параметров ОПУ M, K и KK. Типизация систем осуществляется в подпрограмме TRAM.

Пусть $S(KK,K)$ обозначает общий тип системы (I). Для программы REGN допустимы следующие типы решаемых систем (I).

$S(1,4)$ - Система различающихся между собой уравнений, или система однотипных уравнений, отличающихся между собой по значению одного параметра (в TRAM и RELADI обозначается как $X(1)$).

Параметр $X(1)$ автоматически принимает значение

$$X(1) = 1, 2, \ldots, \bar{M}.$$

$X(1)$- элемент из общего для TRAM и RELADI блока $/X/$. Матрица $G=E$. Управление M имеет значение $\bar{M}$.

Структура входных данных имеет вид

$$YR(1) = y_1$$
$$\vdots$$
$$YR(\bar{M}) = y_{\bar{M}}.$$

$S(2,2)$- Система одинаковых по типу уравнений, различающихся только по значениям одного параметра $X(1)$, который входит во входные данные задачи. Матрица $G=E$. Управление M имеет значение $2\bar{M}$.

Структура данных имеет вид

$$\begin{array}{ll} YR(1)=y_1, & YR(2)=p_1 \\ \vdots & \vdots \\ YR(2\bar{M}-1)= y_{\bar{M}}, & YR(2\bar{M})=p_{\bar{M}}. \end{array}$$

Параметр $X(1)$ принимает значения

$$X(1)=p_1, \ldots, X(\bar{M})=p_{\bar{M}}.$$

$S(3,3)$ - Система уравнений, которая отлична от типа $S(2,2)$ только тем, что $G=\begin{bmatrix} g_1 & & 0 \\ & \ddots & \\ 0 & & g_{\bar{M}} \end{bmatrix}$ и что структура входных данных имеет вид

$$\begin{array}{lll} YR(1)=y_1, & YR(2)=\frac{1}{g_1}, & YR(3)=p_1 \\ \vdots & \vdots & \vdots \\ YR(3\bar{M}-2)= y_{\bar{M}}, & YR(3\bar{M}-1)= \frac{1}{g_{\bar{M}}}, & YR(3\bar{M})=p_{\bar{M}}. \end{array}$$

$S(\overline{KK},\bar{K})$- Система одинаковых (или различных по виду) уравнений, в которых участвуют $\overline{KK}-\bar{K}$ задаваемых во входных данных параметров $X(1),\ldots,X(\overline{KK}-\bar{K})$.

На числа $\overline{KK}$ и $\bar{K}$ наложены ограничения $2 \le \bar{K} \le 4$ и $\bar{K} < \overline{KK} \le 10$.

Матрица $G=\begin{bmatrix} g_1 & & 0 \\ & \ddots & \\ 0 & & g_{\bar{M}} \end{bmatrix}$. Управление M имеет значение $q\bar{M}$, где $q=2+\overline{KK}-\bar{K}$.

Структура входных данных имеет вид

$$
\begin{array}{llll}
YR(1)=y_1, & YR(2)=\dfrac{1}{g_1}, & YR(3)=p_1^{(1)}, \ldots, & YR(q)=p_1^{(q)} \\
\vdots & \vdots & \vdots & \vdots \\
YR(q(\bar{M}-1)+1)= & YR(q(\bar{M}-1)+2)= & YR(q(\bar{M}-1)+3)=, \ldots, & YR(q\bar{M})= \\
=y_{\bar{M}} & =\dfrac{1}{g_{\bar{M}}} & =p_{\bar{M}}^{(1)} & =p_{\bar{M}}^{(q)}
\end{array}
$$

Параметры задачи принимают значения

$$X(1)=p_1^{(1)}, \ldots, p_{\bar{M}}^{(1)},$$

$$\vdots$$

$$X(q)=p_1^{(q)}, \ldots, p_{\bar{M}}^{(q)} .$$

П. 9. Статистическая оценка решения

При решении переопределенных нелинейных систем уравнений ($N<\bar{M}$) на основе R-процессов Гаусса-Ньютона возможна статистическая оценка получаемого решения. Для этой цели к наилучшей итерации применяется теория ошибок метода наименьших квадратов /I6, стр. I92/.

В блоке STATICALANALISIS программы реализована обобщенная оценка решения $x_{\bar{n}}$ ($\bar{n}$ – номер наилучшей итерации), для которой в качестве информационной матрицы берется матрица

$$F(\varepsilon^*)=\frac{HI\,SQ}{\bar{M}-N+K}\left(J^T(x_{\bar{n}})\,G\,J(x_{\bar{n}})+\varepsilon^* U\right)^{-1},$$

где $\varepsilon^*=\begin{cases} 0, & \text{если матрица } (J_{\bar{n}}^T G J_{\bar{n}})^{-1} \text{ существует;} \\ \varepsilon_{\bar{n}}, & \text{если матрица } J_{\bar{n}}^T G J_{\bar{n}} \text{ необратима.} \end{cases}$

K – число заданных крестов для операции сжатия (П. 6).

Статистические ошибки неизвестных $x_{1,\bar{n}}, \ldots, x_{N,\bar{n}}$ и коэффициенты корреляции между ними вычисляются по формулам /I6, стр. 202/.

$$\Delta x_{i,\bar{n}}(\varepsilon^*)=\sqrt{F_{ii}(\varepsilon^*)},$$

$$B_{ij}(\varepsilon^*)=\frac{F_{ij}(\varepsilon^*)}{\sqrt{F_{ii}(\varepsilon^*)\,F_{jj}(\varepsilon^*)}}, \tag{2I}$$

где F_{ij} $(i,j=1,\ldots,N)$ – i-j-тый элемент матрицы F.

В случае, когда $\varepsilon^*=0$, приведенные оценки являются оценками в смысле наименьших квадратов (несмещенными и с минимальной дисперсией), чего нельзя сказать о случае $\varepsilon^*=\varepsilon_{\bar{n}}>0$.

Оценки (2I) при достаточно маленьких числах $\varepsilon^*>0$ можно использовать как носители качественной информации о том, какие из компонент решения $x_{\bar{n}}$ делают матрицу $J_{\bar{n}}^T G J_{\bar{n}}$ сингулярной. Очевидно, таким компонентам будут отвечать большие значения $\Delta x_{i,\bar{n}}(\varepsilon^*)$ и близкие к I значения $B_{ij}(\varepsilon^*)$.

Отметим, что использование оценок (2I) в случае $\varepsilon^*=\varepsilon_{\bar{n}}>0$ хорошо сочетается с характерным для процессов типа ARP стремлением к нулю числа регуляризации ε_n при $n\to\infty$.

В случаях, когда имеют место оценки (21) при $\varepsilon^* > 0$, программа REGN печатает индикацию „QUASI-ERRORS AND QUASI-CORRELATION COEFFICIENTS" .

Используя программу REGN, можно проводить g-обращение (см. П. 6) при помощи операции "сжатие-обращение-растяжение". В таком случае удаляемым крестам должны точно соответствовать те компоненты вектора $x_{\bar{n}}$, которые делают матрицу $J_{\bar{n}}^T G J_{\bar{n}}$ сингулярной. Информационная матрица при этом имеет вид

$$\bar{F}(0) = \frac{HI\ SQ}{\bar{M}-N+k} \uparrow_k \left(\left(\downarrow_k (J_{\bar{n}}^T G J_{\bar{n}}) \right)^{-1} \right) .$$

В нелинейном случае оценки (21) на основе $\bar{F}(0)$ могут быть использованы как оценки для всех компонент вектора $x_{\bar{n}}$ за исключением тех, которые соответствуют удаляемым крестам, только если ошибки в последних являются пренебрежимо малыми.

Примером применимости статистической оценки на основании матрицы $\bar{F}(0)$ является метод нахождения числа скрытых закономерностей/17,18/ (см. также пример 3 в П. 13).

Во многих случаях при обработке экспериментальных данных стандартные отклонения σ_i измеренных величин y_i ($i = 1,\ldots,\bar{M}$) являются хорошо определенными. Тогда в процессах R^{GN} в качестве матрицы взвешивания берется $G = \begin{bmatrix} \frac{1}{\sigma_1^2} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\sigma_{\bar{M}}^2} \end{bmatrix}$. Критерием успешного решения таких задач служит приблизительное равенство $\frac{HI\ SQ}{\bar{M}-N} \approx 1$ при малом значении критерия цели $R\Phi$.

В програме REGN предусмотрена возможность получения оценок $\Delta x_{i,\bar{n}}(\varepsilon^*)$ и $B_{ij}(\varepsilon^*)$ ($i,j = 1,\ldots,N$) в описанном выше общем случае.

Управление приведенным статистическим анализом осуществляется с помощью управления выдачей полученной информации (см. следующий пункт).

П. 10. Выдача полученной информации

При работе с программой REGN найденная информация о решении системы (1), а также информация о характере осуществленного процесса (5) в настоящее время выводится на печатающее устройство. Это производится при помощи специальной подпрограммы OUTPUT.

Выдаваемая информация имеет следующие элементы.

D - печать невязок

$$f_j(x_{\bar{n}}) - y_j \quad (j = 1,\ldots,\bar{M});$$

C - печать критериев процесса (5)

(печатаются также EXTITT - номер итерации в процессе (5) и INTITT - число проведенных внутренних итераций (необходимых для осуществления минимизации (10)) в режиме RPBC);

X - печать компонент вектора x_n ;

XL(C) - печать компонент вектора $x_{\bar{n}}$ на наилучшей итерации в смысле выбранного критерия цели (это та итерация, для которой заданный критерий цели C имеет минимальное значение);

XL+SA - печать компонент вектора $x_{\bar{n}}$ вместе со статистическими оценками (печатаются оценки (21)).

В зависимости от того, к какой итерации относится выдаваемая информация, различаются три типа печати.

PI - начальная печать, относящаяся к начальному приближению x_0;

P - промежуточная печать, которая относится к приближениям x_n $(n=1,\dots,n^*)$;

PF - конечная печать, которая относится к наилучшему решению в смысле выбранного критерия RØ, MAX DEFECT или HI SQ.

В программе REGN реализованы следующие типы комбинированной выдачи (которые, в сущности, составляют модификации класса P).

PI : $(C+X)_0$, $(D+C+X)_0$;

P : $(C+X)_n$, $(D+C+X)_n$;

PF : $(C+XL)_F$, $(D+C+XL)_F$,

$(C+XL+SA)_F$, $(D+C+XL+SA)_F$.

Управление модификациями класса P осуществляется при помощи параметра KP типа ОПУ и параметров KPI, KPF и L типа ВПУ (элементы общего блока KSEL) в соответствии со следующей таблицей.

Таблица 5

ВПУ \ PI	$(C+X)_0$	$(D+C+X)_0$		
KPI	1	2		
ОПУ \ P	$(C+X)_n$	$(D+C+X)_n$		
KP	1	2		
ВПУ \ PF	$(C+XL(C))_F$	$(D+C+XL(C))_F$	$(C+XL(C)+SA)_F$	$(D+C+XL(C)+SA)_F$
KPF	1	2	3	4

Критерий цели C в конечной печати PF задается при помощи параметра L. Значения L = 1, 2, 3 отвечают, соответственно, C = RØ, MAX DEFECT, HI SQ.

Алгоритм, выбирающий минимальное значение C, нужное для конечной печати (см. блок BEST ITERATIØN), использует в качестве большого числа параметр ВПУ C(15). В режиме самоуправления C(15)=1.E+13.

П. II. Составление програмы USER и RELADI

Применение программы REGN для решения одной или нескольких систем уравнений (I) осуществляется при помощи двух дополнительных программ. В первой из них (называемой в дальнейшем USER) задаются рабочие массивы, входные данные задачи и необходимые управления для реализации выбранных модификаций алгоритма. Из программы USER осуществляется обращение к программе REGN. Кроме описанных функций, программа $USER$ может выполнять и другие, не имеющие прямого отношения к использованию $REGN$.

Вторая программа ($SUBROUTINE\ RELADI$) составляется исключительно для применения программы $REGN$. В ней программируются функции $\{f_j\}$ и частные производные $\frac{\partial f_j}{\partial x_i}$ если для них имеются аналитические выражения. Программа $RELADI$ в общем случае реализует магазинную структуру, предназначенную для организации хранения информации, относящейся одновременно к нескольким нелинейным системам уравнений.

Приведем более подробное описание правил составления программ $USER$ и $RELADI$.

При составлении программы $USER$ надо выполнить следующее.

- Зарезервировать массивы $Z(\bar{N},\bar{N})$, $ZL(\bar{N},\bar{NN})$, $YR(\bar{\bar{M}})$ и $XX(\bar{N})$. Размерности $\bar{N}$, $\bar{NN}$, $\bar{M}$ должны быть связаны с параметрами ОПУ, N , NN и $\bar{M}$ соотношениями

$$N \le \bar{N},\quad NN = \bar{NN} \quad \text{и} \quad \bar{M} \le \bar{\bar{M}} .$$

- Заказать общие (с подпрограммой $RELADI$) блоки, которые используются в случае наличия управлений типа ВПУ (блоки $KSEL$, $LINT$, $WEPS$, $WENORM$, WEN , $CINT$, $DROPX$).

- Выбрать типизацию системы и согласно ей задать входные данные задачи, т.е. в массиве YR задать (используя нужную структуру) вектор y , матрицу взвешивания G и параметры задачи (если таковые имеются - см. П. 8).

- В массиве XX заносится начальное приближение (вектор x_0).

- Задать управления (ПУ). В случаях, когда $RELADI$ имеет магазинную структуру, параметр NP служит для задания номера решаемой задачи. При использовании режима самоуправления параметрам ВПУ можно ничего не присваивать.

- Выполнить обращение к программе $REGN$. Для упрощения работы рекомендуется в обращения вместо фактических констант употреблять идентификаторы, которым присваивать значения перед обращением, как это показано в программе $EXAMP$ (см. приложение 2).

После оператора $CALL\ REGN$ в программе $USER$ все заданные управления сохраняются. Найденное решение (в смысле наилучшей итерации) содержится в массиве XX . В общих блоках $CRIT$, $CRITW$ и $EPSW$ содержится информация о критериях цели относительно наилучшей итерации. Типичным примером программы $USER$ может служить программа $EXAMP$ (см. приложение 2).

Составление программы $RELADI$ зависит от типа решаемой системы (I), наличия аналитических выражений для частных производных и от числа решаемых задач.

Приведем одну из возможных схем программы $RELADI$, реализующую магазинную структуру (решаются две задачи) с необходимыми пояснениями.

```
      6
      SUBROUTINE RELADI(N,NP,INDEX,A,Y,DF)
      DIMENSION A(N),DF(N)
      COMMON/X/ X(10)
```

Ставится, если среди решаемых задач имеется задача, зависящая от параметров.

$$GO\ TO\ (101,102),NP$$

Этот переход разбивает программу RELADI на части, отвечающие решаемым задачам. Номер $NP=1$ означает первую задачу и т.д.

```
101  GO TO (11,12,...,10+M̄),INDEX
```

Начало первой части RELADI . Присутствие GO TO с меткой 101 указывает, что задача имеет различные между собой по виду связи f_j. Вместо $x_1,\dots,x_N$ пишутся $A(1),\dots,A(N)$.

```
11      Y = f_1 (x_1,...,x_N)
        DF(1) = ∂f_1/∂x_1
          ⋮
        DF(N) = ∂f_1/∂x_N
12      ______
          ⋮
10+M̄    Y = f_M̄ (x_1,...,x_N)
        DF(1) = ∂f_M̄/∂x_1
          ⋮
        DF(N) = ∂f_M̄/∂x_N
        RETURN
```

Начало второй части программы RELADI

```
102     Y = f (x_1,...,x_N; X(1),X(2),X(3))
        DF(1) = ∂f/∂x_1
          ⋮
        DF(N) = ∂f/∂x_N
```

Вторая задача имеет однотипные связи (обозначенные f), которые различаются между собой только значениями троек ($X(1)$, $X(2)$, $X(3)$) задаваемых параметров (см. П. 8). При численном дифференцировании операторы $DF(1)=\cdots;\cdots;DF(N)=\cdots$ опускаются.

```
        RETURN
        END
```


Примером RELADI , написанной по приведенной схеме, может служить RELADI , которую использует программа EXAMP (см. приложение 2).

Одним из возможных применений программы REGN является построение новых (на более высоком уровне) подпрограмм для решения системы (I), имеющих композиционные алгоритмы, составленных из нескольких видов R -процессов (5).

Примером такого алгоритма может служить алгоритм, составленный из одной итерации RPBC и модификации ARP для продолжения процесса с начальной постоянной ε_0 , равной числу $\varepsilon_1^{Q_s}$, выработанному RPBC . Такая композиция может существенно облегчить применение авторегуляризованных процессов (5) на основе связи (7).

П. I2. Решение линейных систем и задачи о собственных значениях

В программе REGN для решения линейных систем уравнений и полной задачи о собственных значениях для симметрической матрицы предусмотрены некоторые специальные возможности.

Пусть задана линейная система уравнений

$$Zx=y\ ,\ (x\in R^N, y\in R^{\bar M})\ . \tag{22}$$

В случаях, когда матрица Z^TZ плохо обусловлена, решение системы (22) на основе модификаций ERP или ARP итерационного процесса (5) может оказаться выгоднее, чем другими методами.

Решение (22) в REGN осуществляется при помощи специальной подпрограммы RELAD .

При решении (22) матрица $REGN$ задается в массиве $ZL(\bar{N},\bar{NN})$, который должен быть зарезервирован в $USER$ с точными размерностями $\bar{N}=N$ и $\bar{NN}=\bar{M}$. Свободный вектор y задается в YR. В программе $REGN$ при решении (22) реализована только возможность типа $S(1,4)$.

Допустимыми модификациями класса A являются: ARP-F, ARP и ERP.

Основным управлением при применении $REGN$ в линейном случае служит $NP=1$. При счете на ЭВМ CDC (серии 6000) программа $RELADI$ не составляется, а при счете на БЭСМ-6 некоторая программа с именем $RELADI$ необходима, хотя к ней фактически и не происходит обращения. Естественным начальным приближением является нулевой вектор.

Пусть Z - действительная симметричная матрица порядка $N-1$, действующая над элементами $\bar{x}=(x_2,\dots x_N)^T$. Рассматриваем в R^N оператор

$$f x = \begin{bmatrix} Z\bar{x} - x_1\bar{x} \\ \sum\limits_{i=2}^{N} x_i^2 = 1 \end{bmatrix}, \text{где } x = \begin{bmatrix} x_1 \\ \bar{x} \end{bmatrix}$$

и нелинейное уравнение

$$f x = \theta \quad . \tag{23}$$

В этом уравнении компонента x_1 является собственным значением матрицы Z, а вектор $(x_2,\dots,x_N)$ - соответствующий ему собственный вектор. Оператор f и его матрица Якоби запрограммированы в подпрограмме $RELAD$.

При решении уравнения (23) можно использовать все модификации класса A. Допустимая типизация $S(1,4)$. Матрица Z задается в массиве ZL, начиная с элемента $ZL(1,1)$.

Массив $ZL(\bar{N},\bar{NN})$ должен быть зарезервирован в $USER$ с размерностями $\bar{N}=\bar{NN}=N$. Основным управлением при решении (23) является $NP=-2$.

Нелинейная система (23) имеет $N-1$ решений. Программу $EXAMP$ удобнее применять как уточняющую программу вместе с другой программой, которая задает начальные приближения для всех решений (например, программа, основанная на методе вращения Якоби).

П. 13. Примеры применения программы $REGN$

В программе $EXAMP$, которая приводится в приложении 2, даются три примера решения нелинейных систем уравнений с помощью $REGN$.

Эти примеры подобраны так, что вместе охватывают все основные модификации алгоритма программы $REGN$. Таким образом, программу $EXAMP$ можно использовать как пособие при изучении $REGN$.

С другой стороны, решение примеров в программе $EXAMP$ является конкретными образцами составления программы типа $USER$.

Приведем необходимые объяснения и комментарии к этим примерам.

Пример I.

Решается нелинейная система[2]

$$\left.\begin{aligned} x_1^2 + x_2 &= 2 \\ x_1 + x_2^2 &= 0 \end{aligned}\right\} \tag{22}$$

Эта система имеет два действительных решения

$x^*: x_1=-1, x_2=1;$

$x^{**}: x_1 \approx -1.83117721, x_2 \approx -1.35320996.$

Система (22) решается с начальным приближением

$x^\circ: x_1=-0.5, x_2=-0.5$.

Для этого начального приближения матрица Якоби $J_0^T J_0$ вырождается и задачу невозможно решить без применения R -процессов.

Система (22) в программе EXAMP причислена к типу $S(1,4)$. В качестве модификации процесса (5) выбрана ARP-F с начальным числом $\mathcal{E}_0=1$. (О выборе других управлений см. непосредственно в программе EXAMP).

На стр. 68 приводится полученная информация в виде $(C+X)_0 + (C+X)_n$. Решение x^* находится на 6-той итерации с точностью до II-того знака после запятой.

Пример 2.

Система (22) решается с новым начальным приближением

$x^\circ: x_1=-0.5, x_2=-0.4$.

Здесь матрица Якоби $J_0^T J_0$ является только плохо обусловленной /I2/. Применение обыкновенного метода Гаусса-Ньютона приводит к расходимости после первой итерации. В программе EXAMP при решении (22) в этом случае используется численное дифференцирование D2) с шагом $D=0.000000001$.

В качестве основной модификации для решения (22) привлечена RPBC .

Остальные управления сохраняются такими же, как в примере I.

Найденная информация по итерациям приводится на стр. 69 . Как видно из печати, на первой итерации (EXTITT=1) число регуляризации $\mathcal{E}_1^{Qs}=1.81$ в режиме RPBC определяется при помощи 26 внутренних итераций. (LNTITT=26 при управлениях AD=1 и $S=0.1$). На второй итерации, аналогично, число регуляризации $\mathcal{E}_2^{Qs}=0.601$ определяется при помощи 2I внутренних итераций. От третьей до шестой итераций число внутренних итераций равно 2, (минимальное число в режиме RPBC). В качестве чисел $\mathcal{E}_n^{Qs}$ берется константа 0.001 , которая является значением параметра C(14) типа ВПУ (на внутреннем языке EBCL - в режиме самоуправления).

Пример 3.

Этот пример дает иллюстрацию более гибкого использования программы REGN в случае решения одной задачи анализа экспоненциальных зависимостей.

Рассматриваем экспоненциальную зависимость $y(t), t \geqslant 0$ /I9, стр. 280/

$$0.0951 e^{-t} + 0.8607 e^{-3t} + 1.5576 e^{-5t} = y(t) \qquad (24)$$

Пусть значения $y(t_j)$ для сетки $t_1=0, t_{j+1}=t_j+0.05 (j=1,\dots,23)$ вычислены с точностью до I0-того знака после запятой.

Рассматриваем нелинейную систему из 24 уравнений

$$\left.\begin{aligned} &\sum_{i=1}^{4} x_i e^{-x_{4+i} t_j} = y(t_j) \\ &j=1,2,\dots,24 . \end{aligned}\right\} \qquad (25)$$

относительно неизвестных амплитуд $x_1, \dots, x_4$ и неизвестных декрементов $x_5, \dots, x_8$. Полученная система уравнений (25) имеет сингулярную матрицу Якоби в самом решении по причине допущения существования 4-той экспоненты /I7/. Указанная сингулярность, а также сильная нелинейность, делают решение системы (25) довольно нелегкой задачей /I8,I9/.

Решение системы (25) в программе EXAMP разбивается на две части. В первой части (ей отвечает первое обращение к REGN в EXAMP3, см. стр. 66) система (25) решается только для определения числа экспонент. Во второй части найденное решение уточняется новым итерационным процессом, в котором производится g-обращение матриц S_n (см. П.6) и делается статистическая оценка полученного решения.

В первой части система (25) причислена к типу $S(3,3)$. Из этого следует тройная структура данных при задании входной информации $(M=3\bar{M})$. Величины $y(t_j)$ $(j=1,\dots,24)$ не являются скоррелированными, и в этом случае матрица взвешивания G равна единичной. Параметром задачи служит t_j, $j=1,\dots,24$.

Для решения (25) привлекается модификация ARP-F с начальным числом $\varepsilon_0=10$. В качестве модификации прерывания выбрана G3). Полученная информация печатается в виде $(C+X)_0+(C+XL)_F$ (см. стр. 70). В части $(C+X)_0$ этой печати видны начальные приближения и соответствующие им критерии цели. В части $(C+XL)_F$ печати дается 30-тая итерация как наилучшее расширение. Критерий MAX DEFECT показывает, что решение на 30-той итерации восстанавливает заданные функциональные значения $\{y(t_j)\}$ с точностью до 5-того знака после запятой. Большое значение критерия COND подтверждает

ожидаемую плохую обусловленность матрицы $J_{30}^T J_{30}$. Число $\|(J_{30}^T J_{30}+\varepsilon_0 E)^{-1}\| = (COND)_{30}/((TAU)_{30}+(EPS)_{30})$ тоже является относительно большим. Это означает, что итерационный процесс в первой части решения (25) обладает одновременно двумя противоречивыми свойствами - "самоисправление" /I/ и "отталкивание по $(J^T(x)J(x)+\varepsilon(x)E)^{-1}$" /I/, которым объясняется проявленное замедление в сходимости. Успех при решении такого типа задач обусловлен авторегуляризацией привлеченного итерационного процесса. При рассмотрении найденного решения на 30-той итерации можно заметить, что декременты X(7) и X(8) различны между собой только на 3%. Этот факт дает основание сделать гипотезу, что в системе (25) существуют не четыре экспоненты, а только три /I7,I8/. Вторая часть решения системы (25) основывается на этой гипотезе. В качестве амплитуды четвертой экспоненты берется нуль $(XX(4)=0)$ и фиксируется $(DX(4)=1)$. Декремент x_8 тоже фиксируется $(DX(8)=1)$. Проведенные фиксирования означают, что задана информация об удалении 4-того и 8-го крестов из матриц $J_n^T J_n$. Сохраняя все остальные управления теми же, что в первой части решения задачи, производится новый итерационный процесс, на каждой итерации которого осуществляется обращение типа $\updownarrow((\downarrow S_n)^{-1})$. Как легко можно заметить, сделанная выше гипотеза эквивалентна гипотезе, что обращения $S_n^{(2)}$ являются g-обращениями.

Новый итерационный процесс полностью решает поставленную задачу. На стр. 69 дается печать найденной информации в виде $(C+X)_0+(D+C+XL+SA)_F$. Часть $(D+C+XL+SA)_F$ печати относится к I7-той итерации, которая оказалась для процесса наилучшей. С другой стороны, критерий MAX DEFECT на I7-той итерации показывает, что найденное решение восстанавливает заданные значения с точностью до 10-того знака. Это означает, что лучшее решение задачи (25) практически невозможно.

Вторая часть решения (25) имела характер численного эксперимента. Результат этого эксперимента показывает, что сделанная гипотеза была правильной. Большое значение критерия поведения COND в решении задачи показывает, что анализ приведенной экспоненциальной зависимости (24) является трудной задачей (в частности, сильно неустойчивой) даже когда она решается с точным числом экспонент.

Применение программы REGN в примере 3 является в некоторой мере типичным при решении реальных задач из ядерной физики.

Автор выражает глубокую благодарность Д.Караджову, Н.Ангелову, С.Аврамову и Хр.Черневу, многократно оказывавшим ему помощь при работе над программой REGN, а также Г.Л.Мазному за большую помощь в оформлении рукописи и сделанные замечания.

Рукопись поступила в издательский отдел
18 июня 1973 г.

Литература

1. Л.Александров. Сообщение ОИЯИ, Р5-7258, Дубна, 1973.
2. Л.Александров. Сообщение ОИЯИ, Р5-5515, Дубна, 1970.
3. Л.Александров. Сообщение ОИЯИ, Р5-5137, Дубна, 1970.
4. В.Гаджоков. Сообщение ОИЯИ, Р10-5035, Дубна, 1970.
5. В.Гаджоков. В сб. "Труды II школы ОИЯИ - ЭВМ в экспериментальной физике", ОИЯИ, 10-5255, Дубна, 1970.
6. Х.Сакмен. Решение задач системой человек-ЭВМ. Принстон, 1971 (М., 1973).
7. И.Н.Силин. Материалы Совещания по программированию и выч. методам физ. задач, Дубна, май 1969. ОИЯИ, 11-4655, Дубна, 1969, стр. 43-46.
8. SCØPE reference manual 6000 version 3.3. CØNTRØL DATA pub. No. 60305200.
9. С.Н.Соколов, И.Н.Силин. Препринт ОИЯИ, Д-810, Дубна, 1961.
10. Y.Bard. SIAM J. Numer. Anal., v. 7, No. 1, 157-186 (1970).
11. И.Бабушка, Э.Витасек, М.Прагер. Численные процессы решения дифференциальных уравнений, М., 1969.
12. Дж.Форсайт, К.Молер. Численное решение систем нелинейных алгебраических уравнений. М., 1969.
13. Г.Корн, Т.Корн. Справочник по математике. М., 1968.
14. В.В.Галактионов и др. Библиотека программ на ФОРТРАНе (том. I). ОИЯИ, Б1-11-5144, Дубна, 1969.
15. H.Rutishauser. Comm. of the ACM, v. 6, No. 2.
16. С.Р.Рао. Линейные статистические методы и их применения. М., 1968.
17. Л.Александров. ЖВМ и МФ, т. 10, №5, 1285-1287 (1970).
18. Л.Александров. Сообщение ОИЯИ, 5-6821, Дубна, 1972.
19. К.Ланцош. Практические методы прикладного анализа, М., 1961.

ПРИЛОЖЕНИЕ I. ПРОГРАММА REGN НА ФОРТРАНе

```
      SUBROUTINE   REGN
     *(NP,M,N,NN,K,KK,IT,KP,T,AD,S,TT,D,E,Z,ZL,YR,XX,AM,AN)
C
C
      COMMON/A/A(100)/DF/DF(100)/X/X(10)
      COMMON/X1/X1(100)/X2/X2(100)/X3/X3(100)
     */X4/X4(100)/X5/X5(100)/XW/XW(100)
      COMMON/ICON/NC1,NC2,NC3,NC4,NC5,NC6,NC7,NC8,NC9,NC10,
     1NC11,NC12,NC13,NC14,NC15,NAUT
      COMMON/CON/CN1,CN2,CN3,CN4,CN5,CN6,HYMAX/EL/EL
      COMMON/CRIT/ROM,RM,REV,TAU,COND,ENO/EPSW/EPS,EPS1
      COMMON/CRITW/ROMW,RMW,REVW,CD1,EPS1W,TAU1W
      COMMON/KSEL/KPI,KPF,KGS,L/LINT/LINT/NUITT/NUITT/NL/NL
      COMMON/HEPS/HEPS(100)/HENORM/WEN(100)/SIGMA/SIGMA(100)
      COMMON/CINT/C(15)/DROPX/DX(100)
      DIMENSION Z(N,N),ZL(NN,N),XX(N),YR(M)
C
C              *A*
C
C       PREPARATORY SEGMENT
C
C*A/1*          EXTERNAL CONSTANTS
C
      ITT=IT
      KPW2=KPI
      KPW1=KPF
      KPW3=KGS
      T2=TT
      TOLER=T
      KLEF=KK
      KEY=K
      SAD=S
      DERIV=D
      EPS=E
      ALPHA1=AM
      ALPHA2=AN
      NS=NN
      NN=0
C
C*A/2*        INTERNAL CONSTANTS
C
      CEPS=C(1)
      EPSM=C(3)
      TADD=C(4)
      EFALI=C(6)
      BEG=C(7)
      TINV=C(9)
      EPSLOW=C(10)
      EPST=C(11)
      TEPS=C(13)
      EBCL=C(14)
      IF(CEPS) 23,20,23
   20 CEPS=5.
   23 IF(EPSM)25,24,25
   24 EPSM=.1
```



```
   25   IF(TADD)27,26,27
   26 TADD=10.
   27   IF(EFALI)31,28,31
   28   EFALI=0.0001
   31 IF(BEG)33,32,33
   32 BEG=1.
   33 CONTINUE
      IF(TINV) 37,36,37
   36 TINV=1.E+5
   37 CONTINUE
      IF(EPST) 45,44,45
   44 EPST=.5
   45  CONTINUE
      IF(TEPS)49,48,49
   48  TEPS=1.
   49 IF(LINT) 50,50,55
   50 LINT=100
   55 IF(L) 52,52,51
   52 L=2
   51 IF(EBCL)56,56,57
   56 EBCL=0.001
   57 IF(C(15))54,54,53
   54 C(15)=1.E+13
   53 CONTINUE
C
C*A/3*          FORMATION OF TRANSITIONS INDICATIONS
C
      NC1=KLEF
      NC3=N
      IF(DERIV) 1010,1011,1012
 1011  NC4=0
      GO TO 1020
 1010 NC4=1
      DERIV=-D
      GO TO 1020
 1012 NC4=2
 1020  CONTINUE
      IF(ALPHA2) 1050,1050,1051
 1050  NC5=1
      GO TO 7777
 1051  NC5=2
 7777 IF(ALPHA1) 1008,1009,1009
 1008 NC6=1
      ALPHA1=ABSF(ALPHA1)
 1009 CONTINUE
       IF(IT) 1000,1001,1000
 1001  NC8=1
       GO TO 1030
 1000  IF(TOLER)1002,1003,1002
 1003  NC8=2
       GO TO 1030
 1002  NC8=3
 1030  CONTINUE
      IF(SAD) 1005,1005,1004
 1005  IF(AD) 1006,1007,1006
```

```
 1007   NC9=0
        GO TO 1040
 1006   NC9=2
        GO TO 1040
 1004   NC9=1
 1040   CONTINUE
      NC10=NP
        DO 1041 I=1,N
         IF(DX(I)-1.) 1041,1042,1041
 1042   NC11=1
 1041   CONTINUE
      IF(ABSF(D)-100)1066,1067,1067
 1066 NC13=0
        GO TO 1070
 1067 NC13=1
      CN6=ABSF(D)-100
 1070   NAUT=KEY
        LL=M
        M=M/KLEF
C
C*A/4*           PREPARATION FOR ITERATIONS
C
        DO 383 K=1,N
      A(K)=XX(K)
 383    CONTINUE
      IF(NC10+2)1063,1064,1063
 1064 DO 1065 I=1,M
 1065 YR(I)=0.
 1063 CONTINUE
        DO 2006 I=1,N
        IF(WEPS(I))2006,2007,2006
 2007    WEPS(I)=1.
 2006 CONTINUE
        DO 2 I=1,N
        IF(WEN(I))2,2222,2
 2222    WEN(I)=1.
   2    CONTINUE
        NUITT=0
        CALL TRAM (LL,KP,M,N,D,Z,ZL,YR)
          TAX=ANORZF(N,Z)
          ROX=AMAXF(N,X1)
      RM=WYMAX
       RO=ROX
      TAU=TAX
         IF(ROX-1.E-10)1991,1991,1992
 1991    PRINT 4999
         GO TO 1160
 1992    CONTINUE
      NUIT=0
        IF (EPS)1998,1999,1999
 1998    EPS=EPSM*TAX
       AD=EPS
 1999    CONTINUE
C                 *B*
C
```



```
C         FIRST ITERATION  ARP  AND  RPBC
C
          IF(IT) 5900,5600,5600
 5900       CONTINUE
        CALL OUTPUT
       *(KPW2,M,N,0,0,RO,RM,REV,0.,EPS,TAU,Z,XX)
 5555     CONTINUE
         DO 5004 I=1,N
         Z(I,I)=Z(I,I)+WEPS(I)*(EPS+EPSLOW)
 5004      CONTINUE
         INOIT=1
         GO TO 600
 801     CONTINUE
         ITCOND=1
         GO TO 750
  851     CONDIC=COND
         CALL SYMIN (Z,N,NN,IFAIL,X2,X3,X4)
         ITCOND=2
         GO TO 750
  852    COND=COND*CONDIC
         GO TO 700
  811     CONTINUE
          IF(IFAIL-1) 6100,6110,6100
 6110     IF(KP) 6201,6201,6510
 6510      PRINT  5000
 6201       CONTINUE
           EPS=CEPS*(EPS+EFALI)
         GO TO 5555
 6100    CONTINUE
         BO=ANORZF(N,Z)
         DO 5006 J=1,N
         W10=0.
         DO 5005 I=1,N
 5005    W10=W10+Z(J,I)*X1(I)
         X2(J)=W10*ABSF(1.-DX(J))
 5006     A(J)=A(J)-X2(J)
C
C               *C*
C            CONNECTING SEGMENT
C
         ITT=IT
         IT=-IT
 5620    CONTINUE
         CALL TRAM (LL,KP,M,N,D,Z,ZL,YR)
          TAU=ANORZF(N,Z)
          RO=AMAXF(N,X1)
         RM=WYMAX
         DO 5901 K=1,N
 5901  XX(K)=A(K)
 5600       CONTINUE
  504    CONTINUE
            ENO=ALPHA1*EPS*(EPS+TAX)
          ENO=ENO/ROX
        IF(NC9-1) 500,501,500
 501      CONTINUE
```

```
      CALL OUTPUT(KPH2,M,N,0,0,RO ,RM,REV,0.,EPS,TAU,Z,XX)
      GO TO 503
 500    CONTINUE
        IF(ITT) 502,503,503
  502 CALL OUTPUT(KP,M,N,1,0,RO,RM,REV,COND,EPS,TAU,Z,A)
      IFAIL=0
       NUIT=1
  503 CONTINUE
       ROX5=C(15)
       INDIT=2
       NUITT=0
      GO TO 3333
C
C             *D*
C*BASIS ITERATIONAL CYCLE ERP ,ARP AND RPBC*
C
C
C*D/ERP+ARP+RPBC/1*
C
    3  CONTINUE
       DO 388 K=1,N
       XX(K)=A(K)
 388     CONTINUE
 3333 CONTINUE
      IF(TAU)3334,241,3334
  241 TAU=TAU+1.E-8
 3334 NUIT=NUIT+1
C
C*D/ERP*
C
        GO TO (40,41),NC5
   40 EPS=ALPHA1*EXPF(ALPHA2*(NUIT-1))
        GO TO 42
C*END*D/ERP*
C
C*D/ARP*
C
  41  EPS=1.+(4.*ENO*RO/TAU)/TAU
      EPS=SQRTF(ABSF(EPS))
      EPS=(EPS-1.)*TAU*ALPHA2/2.
 42     CONTINUE
C*END*D/ARP*
C
C*D/RPBC/1*      RP  WITH BEST CORRECTIONS
C
        IF(NC9-1) 350,360,350
  360  CONTINUE
       IF(NP+1) 887,889,887
  887    CONTINUE
       DO 888 I=1,N
       DO 888 J=1,N
 888   ZL(I,J)=Z(I,J)
  889  NUITT=0
       ADD=AD
       EPS=-ADD+EBCL
```



```
      DO 371 K=1,N
  371 XH(K)=A(K)
 358  RR=C(15)
 351   CONTINUE
      EPS1=EPS
      CD1=COND
      NUITT=NUITT+1
      EPS=EPS+ADD+EPSLOW
      DO 370 K=1,N
      XX(K)=A(K)
  370 A(K)=XH(K)
      IF(NP+1) 996,997,996
  997 CALL TRAM (LL,KP,M,N,D,Z,ZL,YR)
      GO TO 350
  996  CONTINUE
      DO 999 I=1,N
      DO 999 J=1,N
 999  Z(I,J)=ZL(I,J)
 350   CONTINUE
C*END*D/RPBC/1*
C
CD/ERP+ARP+RPBC/2*    BASIS SEGMENT
C
      DO 4 I=1,N
       Z(I,I)=Z(I,I)+WEPS(I)*(EPS+EPSLOW)
    4 CONTINUE
C
      GO TO 600
  802  CONTINUE
      ITCOND=3
      GO TO 750
  853  CONDIC=COND
      CALL SYMIN (Z,N,NN,IFAIL,X2,X3,X4)
      ITCOND=4
      GO TO 750
  854  COND=COND*CONDIC
      GO TO 700
  812  CONTINUE
       IF(IFAIL-1) 100,106,100
  106 PRINT 5000
     EPS=CEPS*(EPS+EFALI)
        IF(EPS-1.E+7) 101,100,100
  101 IF(NC9-1)110,351,110
  110 NTV=NUITT
      NUITT=0
      CALL TRAM (LL,KP,M,N,D,Z,ZL,YR)
      NUITT=NTV
      NUIT=NUIT+1
      GO TO 350
  100  CONTINUE
   11 ROM=RO
C               BASIS RELATION
C
      CALL REML(M,N,NUIT)
     DO 6 J=1,N
```

```
      W10=0.
      DO 5 I=1,N
    5  W10=W10+Z(J,I)*X1(I)
       X2(J)=W10*ABSF(1.-DX(J))
    6   A(J)=A(J)-X2(J)
      IF(NC6-1)1054,1053,1054
 1053 DO 7 J=1,N
      W10=0.
      DO 8 I=1,N
    8 W10=W10+ZL(J,I)*X2(I)
      X3(J)=W10*ABSF(1.-DX(J))*WEPS(J)*(EPS+EPSLOW)
    7 A(J)=A(J)+X3(J)
 1054  CONTINUE
C*END*D/ERP+ARP+RPBC/3*
C
C*D/RPBC/2
C
        IF(NC9-1)301,302,301
 302   NL=1
       IF(NUITT-LINT)490,490,300
490    CONTINUE
        RO1=RO
       RM1=RM
       REV1=REV
       CALL TRAM (LL,KP,M,N,D,Z,ZL,YR)
       RO=AMAXF(N,X1)
      RM=WYMAX
      IF(T2)410,355,355
  410 IF(RR-REV)354,354,420
  420     RR=REV
       GO TO 351
 355      IF(RR-WYMAX) 354,354,356
 356    RR=WYMAX
        GO TO 351
 354      DO 385 K=1,N
      AB=ABSF(XX(K)+1.E-10)/100.
      IF(ABSF(XX(K)-A(K))/AB-ABSF(T2))385,385,353
 385      CONTINUE
       GO TO 300
  353 CONTINUE
       EPS=EPS-2.*ADD
         ADD=ADD*SAD
        IF(EPS) 480,386,386
  480   EPS=-ADD+ADD/TADD
        PRINT   5000
 386      CONTINUE
       GO TO 358
  300  CALL OUTPUT
     *(KP,M,N,NUIT,NUITT,RO1,RM1,REV1,CD1,EPS1,TAU,Z,XX)
      DO 382 K=1,N
  382 A(K)=XX(K)
       NUITT=0
 301    NL=0
C*END*D/RPBC/2*
```



```
C
C*D/ERP+ARP+RPBC/3*   BEST ITERATION
C
       REVS=REV
       CALL TRAM (LL,KP,M,N,D,Z,ZL,YR)
      TAU=ANORZF(N,Z)
      RO=AMAXF(N,X1)
      RM=WYMAX
       ROM=RO
      IF(NC9-1) 1100,1131,1100
 1100 CONTINUE
      CALL OUTPUT
     *(KP,M,N,NUIT,D,ROM,RM,REV,COND,EPS,TAU,Z,A)
 1131 CONTINUE
       GO TO(71,72,73),L
 71    R5=ROM
       GO TO 74
 72    R5=RM
       GO TO 74
 73    R5=REV
 74    IF(ROX5-R5) 1120,1120,1130
 1130 ROX5=R5
      DO 1140 K=1,N
      X5(K)=A(K)
 1140  XW(K)=XX(K)
      NUITW=NUIT
       NTW=NUITT
      ROMW=ROM
      RMW=RM
       REVW=REV
       REVSW=REVS
      CONDW=COND
      EPS1W=EPS
      TAU1W=TAU
       GO TO 1110
C
C               *E*
C
C       GOING FROM BASIS ITERATIONAL CYCLE
C
 1120   CONTINUE
       IF(KPW3-1)1110,591,1110
 1110  CONTINUE
      NN5=1
      GO TO (7001,7010,7001), NC8
 7001 CONTINUE
       IF(TOLER) 401,7002,7030
 7030   CONTINUE
      DO 387 K=1,N
       AC=ABSF(XX(K)+1.E-10)/100.
       IF(ABSF(XX(K)-A(K))/AC-TOLER) 387,387,7003
 387     CONTINUE
       NN5=2
       GO TO 7003
 401  CONTINUE
```

```
      IF(R5-ABSF(TOLER))7002,7003,7003
 7002 NN5=2
 7003 GO TO (7020,7010,7010), NC8
 7010     IF(NUIT-IT)7020,7011,7011
 7011   NN5=2
 7020 ROM1=ROM
        GO TO (3,591), NN5
  591 CONTINUE
C
C           STATICAL ANALYSIS
C
      IF(KPH1-3)520,521,521
  521 IF(M-N)520,520,530
  530 CONTINUE
      DO 529 I=1,N
  529 A(I)=X5(I)
      IFAIL=0
  532 NUITT=0
      CALL TRAM(LL,KPH1,M,N,D,Z,ZL,YR)
      DO 531 I=1,N
  531 Z(I,I)=Z(I,I)+EPS1H*IFAIL
      INDIT=3
      GO TO 600
  803 CALL SYMIN (Z,N,NN,IFAIL,X2,X3,X4)
      GO TO 700
  813 IF(IFAIL-1)527,526,527
  526 PRINT 5001
      KPH1=3
      GO TO 532
  527 CONTINUE
      ROS=SQRTF(REVSH/(M-N+NN))
      DO 525 I=1,N
      SIGMA(I)=SQRTF(Z(I,I))*ROS
       IF(DX(I)-1.) 525,515,525
  515  SIGMA(I)=0.
  525  CONTINUE
      DO 522 I=1,N
      DO 523 J=1,N
  523 X4(J)=Z(I,J)/SQRTF(Z(I,I)*Z(J,J))
      DO 524 J=1,N
      IF(I-J) 518,524,518
  518   CONTINUE
      Z(I,J)=X4(J)
  524  CONTINUE
  522 CONTINUE
      DO 519 I=1,N
  519  Z(I,I)=1.
  520 CONTINUE
      CALL OUTPUT
     *(KPH1,M,N,NUITH,NTH,ROMH,RMH,REVH,CONDH,EPS1H,TAU1H,Z,X5)
      DO 1151 K=1,N
 1151 XX(K)=X5(K)
 1160 CONTINUE
       M=LL
      N=NC3
```



```
      NN=NS
       IT=ITT
      K=KEY
      RETURN
C
C                  * F *
C
C     BREAKING OUT OF UNKNOWNS ACCORDING TO THE /DROPX/
C
C                  * CRINGE Z *
C
  600  CONTINUE
      IF(NC11-1)601,602,601
  602  NN=0
      DO 606 I=1,N
      IF(DX(I)-1.) 606,603,606
 603  NN=NN+1
      I3=N-NN+1
      I2=I-NN+1
      DO604 I1=I2,I3
      DO 605 I4=1,N
 605  Z(I1,I4)=Z(I1+1,I4)
      DO 604 I4=1,N
 604  Z(I4,I1)=Z(I4,I1+1)
 606  CONTINUE
      N=N-NN
  601 GO TO (801,802,803),INDIT
C
C                  * ENLARGE Z *
C
  700  CONTINUE
      IF(NC11-1)701,702,701
  702  N=N+NN
      II=0
      DO 730 I=1,N
      N1=N-I+1
      IF(DX(N1)-1.) 715,730,715
  715  II=II+1
      N2=N-NN-II+1
      DO 716 J=1,N
  716 Z(N1,J)=Z(N2,J)
  730  CONTINUE
      DO 703 I=1,N
      IF(DX(I)-1.)707,703,707
  707  DO 708 J=1,N
  708  X4(J)=Z(I,J)
      JJ=0
      DO 704 J=1,N
     IF(DX(J)-1.) 705,713,705
  705  JJ=JJ+1
      X3(J)=X4(JJ)
      GO TO 704
  713  X3(J)=0.
  704  CONTINUE
      DO 710 J=1,N
```

```
  710   Z(I,J)=X3(J)
  703   CONTINUE
       DO 709 I=1,N
       IF(DX(I)-1.)709,712,709
  712   Z(I,I)=TINV
  709   CONTINUE
  701  GO TO (811,812,813),INOIT
C
C              * NUMBER CONDITION *
C
  750   CONTINUE
       DO 777 J=1,N
       W5=0.
       DO 776 I=1,N
  776   W5=W5+ABSF(Z(J,I))/WEN(I)
  777   X3(J)=W5*WEN(J)
       COND=AMAXF(N,X3)
       GO TO (851,852,853,854),ITCOND
C
 4999   FORMAT(14X,14HXO IS SOLUTION)
 5000   FORMAT(/14X,19HCORRECTED ITERATION)
 5001  FORMAT(///14X,56HSOLUTION WITH QUASI-ERRORS
     *AND -CORRELATION COEFFICIENTS)
      END
```

```
        SUBROUTINE  REML (M,N,NUIT)
C
C
      COMMON/DROPX/DX(100)/RM/EA,ED,EF
C
      IF(EA) 1,2,1
    1  DO 3 I=1,N
       IF(DX(I)-1.) 4,3,4
    4 DX(I)=EA*EXPF(-ED*NUIT)+EF
      DX(I)=1.-DX(I)
    3 CONTINUE
    2 CONTINUE
      RETURN
      END
```



```
      SUBROUTINE TRAM (LL,KP,M,N,D,Z,ZL,YR)
C
C
C           FORMING OF JCOBI MATRIX AND
C         GETTING THE PRODUCTS  Z=ZT*Z , X1=ZT*Y
C
      COMMON/A/A(100)/DF/DF(100)/X/X(10)
      COMMON/CON/CN1,CN2,CN3,CN4,CN5,CN6,HYMAX
     COMMON/ICON/NC1,NC2,NC3,NC4,NC5,NC6,NC7,NC8,NC9,NC10,
    INC11,NC12,NC13,NC14,NC15,NAUT
     COMMON/X1/X1(100)/X2/X2(100)/X3/X3(100)
    */X4/X4(100)/X5/X5(100)
     COMMON/CRIT/ROM,RM,REV,TAU,COND,ENO
      COMMON/SIGMA/SIGMA(100)
      COMMON/NL/NL/NUITT/NUITT/NEWT/NT
      DIMENSION Z(N,N),YR(LL),A1(100),S(5)
     EQUIVALENCE(A,A1)
C
C
      IF(KP) 3000,3001,3000
 3000  IF(MOD(KP,2)) 3001,3002,3001
 3002  PRINT 3006
      PRINT 3003
 3001  CONTINUE
      KPM=0
      REV=0.
      P=1.
      Q=1.
      DO 70 I=1,N
       IF(NL) 121,121,122
  121  X1(I)=0.
  122  CONTINUE
      DO 70 J=1,N
   70  Z(I,J)=0.
      HYMAX=0.
      DO 2 I=1,M
      NC2=I
       GO TO(10,11,12,13),NAUT
   10 K5=0
      GO TO 50
   11 K5=1
      GO TO 50
   12 K5=2
      GO TO 50
   13    X(1)=FLOAT(I)
      GO TO 60
   50 K1=NC1-K5
       IF(K1-1) 60,51,51
   51  CONTINUE
      DO 3 K2=1,K1
      K3=NC1*(I-1)+K2+K5
    3  X(K2)=YR(K3)
   60 Y=0.
       IF(NC10)200,210,210
  200 NC12=-NC10
```

```
      CALL RELAD(NC12,LL,M,N,NC2,A1,Y,ZL,DF)
        YY=Y
       GO TO 1
 210    CALL RELADI(N,NC10,NC2,A1,Y,DF)
        YY=Y
C
C             NUMER9CAL DIFFERENTIATION
C
       IF(NUITT) 40,40,1
   40 IF(NC4)30,1,30
 30     CONTINUE
       IF(NC4-1) 1,21,21
  21    DO 20 I1=1,NC3
      IF(NC13)22,22,23
   23   D=CN6*ABSF(A(I1))
   22 CONTINUE
      IF(NC4-1)24,24,25
   24 A(I1)=A(I1)+D
      CALL RELADI(N,NC10,NC2,A1,Y,DF)
      YYY=Y
      A(I1)=A(I1)-2.*D
      CALL RELADI(N,NC10,NC2,A1,Y,DF)
      Y1=Y
      A(I1)=A(I1)-D
       CALL RELADI(N,NC10,NC2,A1,Y,DF)
      Y2=Y
      A(I1)=A(I1)-D
       CALL RELADI(N,NC10,NC2,A1,Y,DF)
        Y3=Y
       DF(I1)=(3.*YYY+10.*YY-18.*Y1+6.*Y2-Y3)/(12.*D)
      A(I1)=A(I1)+3.*D
      GO TO 20
 25     CONTINUE
        A(I1)=A(I1)+D
        CALL RELADI(N,NC10,NC2,A1,Y,DF)
       DF(I1)=(Y-YY)/D
      A(I1)=A(I1)-D
 20     CONTINUE
C                *END NUM .DIFF.*
C
   1    Y=YY
      K4=NC1*(I-1)+1
       GO TO (71,72,73,74),NAUT
   71 Y=Y-CN1
      GO TO 80
 72     Y=Y-YR(K4)
      GO TO 80
 73     Y=(Y-YR(K4))/YR(K4+1)
       P=YR(K4+1)
        Q=SQRTF(P)
      GO TO 80
 74     Y=Y-YR(I)
   80 AB=ABSF(Y)
         IF(AB-1.E+6) 82,82,81
  81    REV=1.E+13
```



```
      GO TO 83
  82  REV=REV+(Q*Y)**2
  83  IF(KP) 3390,3300,3390
C
C             *PRINT D*
C
 3390   CONTINUE
C
      IF(MOD(KP,2)) 3300,3333,3300
 3333  KPM=KPM+1
      S(KPM)=Y
      IF(KPM-4) 3334,3335,3335
 3335 J1=I-3
      J2=I-2
      J3=I-1
      J4=I
      PRINT 3350,J1,S(1),J2,S(2),J3,S(3),J4,S(4)
      KPM=0
      GO TO 3300
 3334  IF(I-M) 3300,3340,3340
 3340  J1=I-KPM+1
      J2=J1+1
      J3=J2+1
      GO TO(3341,3342,3343),KPM
 3341  PRINT 3351,J1,S(1)
      GO TO 3300
 3342  PRINT 3352,J1,S(1),J2,S(2)
      GO TO 3300
 3343  PRINT 3353,J1,S(1),J2,S(2),J3,S(3)
 3300  CONTINUE
C             *END PRINT D*
     IF(WYMAX-AB)100,100,110
  100 WYMAX=AB
  110 DO 90 J=1,N
       IF(NL) 120,120,130
  120 IF(NT)92,93,92
   92 X1(J)=Y
      GO TO 130
   93 X1(J)=X1(J)+Y*DF(J)
 130  IF(NC10+1) 132,131,132
  132  IF(NUITT) 131,131,90
  131 CONTINUE
      DO 91 K=1,N
       IF(NT)96,95,96
   96 Z(J,K)=DF(K)
      GOTO 91
   95 Z(J,K)=Z(J,K)+DF(J)*DF(K)/P
   91 CONTINUE
 90     CONTINUE
    2 CONTINUE
        RETURN
 3003  FORMAT(8X,7HOEFECTS)
 3006  FORMAT(//)
 3350  FORMAT(1X,2HD(,I2,2H)=,E17.10,1X,
     *2HD(,I2,2H)=,E17.10,1X,
```

```
      *2HD(,I2,2H)=,E17.10,1X,
      *2HD(,I2,2H)=,E17.10)
 3351  FORMAT(1X,2HD(,I2,2H)=,E17.10)
 3352  FORMAT(1X,2HD(,I2,2H)=,E17.10,1X,
      *2HD(,I2,2H)=,E17.10)
 3353  FORMAT(1X,2HD(,I2,2H)=,E17.10,1X,
      *2HD(,I2,2H)=,E17.10,1X,
      *2HD(,I2,2H)=,E17.10)
       END
```

```
       SUBROUTINE RELAD (NC12,LL,M,N,INDEX,A,Y,ZL,DF)
C
C
       DIMENSION ZL(LL,N),A(N),DF(N)
      GO TO (201,202),NC12
C
C*LINEAR PROBLEM*
C
  201 CONTINUE
      W=0.
      DO 1 I=1,N
      W=W+ZL(INDEX,I)*A(I)
  1    DF(I)=ZL(INDEX,I)
      Y=W
      RETURN
C
C*EIGEN VALUE PROBLEM*
C
  202 CONTINUE
       IF(INDEX-N) 10,20,20
 10     Y=0.
         DO 8 I=2,N
       DF(I)=ZL(INDEX,I-1)
  8       Y=Y+DF(I)*A(I)
       Y=Y-A(1)*A(INDEX+1)
       DF(INDEX+1)=DF(INDEX+1)-A(1)
         DF(1)=-A(INDEX+1)
      RETURN
   20 Y=0.
       DO 30 I=2,N
       Y=Y+A(I)**2
 30    DF(I)=2.*A(I)
       Y=Y-1.
       DF(1)=0.
      RETURN
      END
```



```
       SUBROUTINE SYMIN (A,N,NN,IFAIL,P,Q,R)
C
       DIMENSION A(1),P(1),Q(1),R(1)
       INTEGER R
C
      IFAIL=0
      EPSILN=1.0 E-12
      EPSIL=1.E-10
       II=N
       K1=N+NN
       DO 200 I=2,N
       K=(I-1)*K1
       DO 200 J=1,N
       KK=K+J
       II=II+1
 200   A(II)=A(KK)
      K=0
      DO 10 I=1,N
   10 R(I)=1.
      DO 65 I=1,N
       IF(A(1)-EPSIL)100,100,201
  201  BIG=0.
      JJ=1
      DO 38 J=1,N
      TEST=ABSF(A(JJ))
      IF(TEST-BIG)37,37,31
   31 IF(R(J))100,37,32
   32 BIG=TEST
      K=J
      KK=JJ
   37 JJ=JJ+N+1
   38 CONTINUE
      IF(K) 100,100,39
  39   IF(ABSF(A(KK)/A(1))-EPSILN)100,40,40
   40 R(K)=0.
      Q(K)=1./A(KK)
      P(K)=1.
      A(KK)=0.0
      KP1=K+1
      KM1=K-1
      IF(KM1)100,50,41
   41 JK=1+N*(K-1)
      DO 49 J=1,KM1
      P(J)=A(JK)
      Q(J)=A(JK)*Q(K)
      IF(R(J))100,48,42
   42 Q(J)=-Q(J)
   48 A(JK)=0.0
   49 JK=JK+1
   50 IF(K-N)51,60,100
   51 KJ=K*(N+1)
      DO 59 J=KP1,N
      P(J)=A(KJ)
      Q(J)=-A(KJ)*Q(K)
      IF(R(J))100,52,58
```

```
52 P(J)=-P(J)
58 A(KJ)=0.0
59 KJ=KJ+N
60 IJK=1
   DO 65 J=1,N
   JK=IJK
   DO 64 K=J,N
   A(JK)=A(JK)+P(J)*Q(K)
64 JK=JK+N
65 IJK=IJK+N+1
   IKJ=N+1
   DO 72 IJK=2,N
   IJKM2=IJK-2
   K=IKJ+IJKM2
   JK=IJK
   DO 70 KJ=IKJ,K
   A(JK)=A(KJ)
70 JK=JK+N
72 IKJ=IKJ+N
    GO TO 101
100 IFAIL=1
101  CONTINUE
    II=N**2+1
    K1=N+NN
    DO 300 I=2,N
    K=(N-I+2)*K1-NN+1
    DO 300 J=1,N
    KK=K-J
    II=II-1
300 A(KK)=A(II)
    RETURN
    END
```



```
      SUBROUTINE OUTPUT
     *(KP,M,N,NUIT,NUITT,ROM,RM,REV,COND,EPS,TAU,Z,XX)
      COMMON/SIGMA/SIGMA(100)/X1/X1(100)/X2/X2(100)
      COMMON/DROPX/DX(100)
      COMMON/DF/DF(100)
      DIMENSION XX(N),Z(N,N)
C
C*PRINT XX*
C
      IF(KP) 1000,1000,2000
 1000  RETURN
 2000  CONTINUE
      PRINT 1
      PRINT 2,NUIT,NUITT,ROM,RM,REV,TAU,COND,EPS
       PRINT 3
       IF(KP-2) 100,100,200
 100  I1=N/4
      K=MOD(N,4)+1
      IF(I1-1) 30,20,20
 20   DO 40 I=1,I1
      J1=(I-1)*4+1
      J2=J1+1
      J3=J2+1
      J4=J3+1
 40   PRINT 10 ,J1,XX(J1),J2,XX(J2),J3,XX(J3),J4,XX(J4)
 30   J1=N-K+2
      J2=J1+1
      J3=J2+1
      GO TO(110,120,130,140),K
 120  PRINT 11,J1,XX(J1)
      GO TO 110
 130  PRINT 12,J1,XX(J1),J2,XX(J2)
      GO TO 110
 140  PRINT 13,J1,XX(J1),J2,XX(J2),J3,XX(J3)
 110      RETURN
C*END PRINT XX*
C
C*PRINT XX,B*
C
 200  DO 201 I=1,N
     IF(DX(I)-1.)202,203,202
 202 CONTINUE
      IF(N-10) 300,300,400
 300  GO TO(301,302,303,304,305,307,307,307,307,307,307),N
 301  PRINT 31,I,XX(I),Z(I,1)
      GO TO 3130
 302  PRINT 32,I,XX(I),Z(I,1),Z(I,2)
      GO TO 3130
 303  PRINT 33,I,XX(I),Z(I,1),Z(I,2),Z(I,3)
      GO TO 3130
 304  PRINT 34,I,XX(I),Z(I,1),Z(I,2),Z(I,3),Z(I,4)
      GO TO 3130
 305  PRINT 35,I,XX(I),Z(I,1),Z(I,2),Z(I,3),Z(I,4),Z(I,5)
 3130 PRINT 60,SIGMA(I)
      GO TO 201
```

```
307    PRINT 36,I,XX(I),Z(I,1),Z(I,2),Z(I,3),Z(I,4),Z(I,5)
       GO TO(311,311,311,311,311,311,312,313,314,315),N
311    PRINT 41,SIGMA(I),Z(I,6)
       GO TO 201
312    PRINT 42,SIGMA(I),Z(I,6),Z(I,7)
       GO TO 201
313    PRINT 43,SIGMA(I),Z(I,6),Z(I,7),Z(I,8)
       GO TO 201
314    PRINT 44,SIGMA(I),Z(I,6),Z(I,7),Z(I,8),Z(I,9)
       GO TO 201
315    PRINT 45,SIGMA(I),Z(I,6),Z(I,7),Z(I,8),Z(I,9),Z(I,10)
       GO TO 201
400    PRINT 36,I,XX(I),Z(I,1),Z(I,2),Z(I,3),Z(I,4),Z(I,5)
       PRINT 46,SIGMA(I),Z(I,6),Z(I,7),Z(I,8),Z(I,9),Z(I,10)
       I1=N-10
       K=MOD(I1,5)+1
       I1=I1/5
       IF(I1-1) 403,402,402
402    DO 404 J=1,I1
       J1=(J-1)*5+11
       J2=J1+1
       J3=J2+1
       J4=J3+1
       J5=J4+1
404    PRINT 410,J1,Z(I,J1),J2,Z(I,J2),J3,Z(I,J3),
     *J4,Z(I,J4),J5,Z(I,J5)
403    J1=N-K+2
       J2=J1+1
       J3=J2+1
       J4=J3+1
       GO TO(201,411,412,413,414),K
411    PRINT 431,J1,Z(I,J1)
       GO TO 201
412    PRINT 432,J1,Z(I,J1),J2,Z(I,J2)
       GO TO 201
413    PRINT 433,J1,Z(I,J1),J2,Z(I,J2),J3,Z(I,J3)
       GO TO 201
414    PRINT 434,J1,Z(I,J1),J2,Z(I,J2),J3,Z(I,J3),J4,Z(I,J4)
      GO TO 201
203   PRINT 204,I,XX(I)
       PRINT 60,SIGMA(I)
201    CONTINUE
         RETURN
204    FORMAT(1X,2HX(,I2,2H)=,E17.10,14X,19H IS DROPPED UNKNOWN)
 1    FORMAT(1X,13HEXTITT INTITT,6X,2HRO, 8X,3HMAX,
     *1X,6HDEFECT,7X,
     15HHI SQ,10X,3HTAU,11X,4HCOND, 9X,3HEPS)
  2    FORMAT(1X,I5,1X,I5,2X,6E14.7)
3      FORMAT(8X,8HUNKNOWNS)
 10    FORMAT(1X,2HX(,I2,2H)=,E17.10,1X,
     *2HX(,I2,2H)=,E17.10,1X,
     *2HX(,I2,2H)=,E17.10,1X,
     *2HX(,I2,2H)=,E17.10)
 11    FORMAT(1X,2HX(,I2,2H)=,E17.10)
 12    FORMAT(1X,2HX(,I2,2H)=,E17.10,1X,
```



```
     *2HX(,I2,2H)=,E17.10)
 13    FORMAT(1X,2HX(,I2,2H)=,E17.10,1X,
     *2HX(,I2,2H)=,E17.10,1X,
     *2HX(,I2,2H)=,E17.10)
 14    FORMAT(1X,2HX(,I2,2H)=,E17.10,1X,
     *2HX(,I2,2H)=,E17.10,1X,
     *2HX(,I2,2H)=,E17.10,1X,
     *2HX(,I2,2H)=,E17.10)
 31    FORMAT(1X,2HX(,I2,2H)=,E17.10,1X,
     *6HB( 1)=,F8.5)
 32    FORMAT(1X,2HX(,I2,2H)=,E17.10,1X,
     *6HB( 1)=,F8.5,1X,6HB( 2)=,F8.5)
 33    FORMAT(1X,2HX(,I2,2H)=,E17.10,1X,
     *6HB( 1)=,F8.5,1X,6HB( 2)=,F8.5,1X,6HB( 3)=,F8.5)
 34    FORMAT(1X,2HX(,I2,2H)=,E17.10,1X,
     *6HB( 1)=,F8.5,1X,6HB( 2)=,F8.5,1X,6HB( 3)=,F8.5,1X,
     *6HB( 4)=,F8.5)
 35    FORMAT(1X,2HX(,I2,2H)=,E17.10,1X,
     *6HB( 1)=,F8.5,1X,6HB( 2)=,F8.5,1X,6HB( 3)=,F8.5,1X,
     *6HB( 4)=,F8.5,1X,
     *6HB( 5)=,F8.5)
 36    FORMAT(1X,2HX(,I2,2H)=,E17.10,1X,
     *6HB( 1)=,F8.5,1X,6HB( 2)=,F8.5,1X,6HB( 3)=,F8.5,1X,
     *6HB( 4)=,F8.5,1X,
     *6HB( 5)=,F8.5)
 60    FORMAT(1X,6H +/-  ,E17.10)
 41    FORMAT(1X,6H +/-  ,E17.10,1X,
     *6HB( 6)=,F8.5)
 42    FORMAT(1X,6H +/-  ,E17.10,1X,
     *6HB( 6)=,F8.5,1X,6HB( 7)=,F8.5)
 43    FORMAT(1X,6H +/-  ,E17.10,1X,
     *6HB( 6)=,F8.5,1X,6HB( 7)=,F8.5,1X,6HB( 8)=,F8.5)
 44    FORMAT(1X,6H +/-  ,E17.10,1X,
     *6HB( 6)=,F8.5,1X,6HB( 7)=,F8.5,1X,6HB( 8)=,F8.5,1X,
     *6HB( 9)=,F8.5)
 45    FORMAT(1X,6H +/-  ,E17.10,1X,
     *6HB( 6)=,F8.5,1X,6HB( 7)=,F8.5,1X,6HB( 8)=,F8.5,1X,
     *6HB( 9)=,F8.5,1X,
     *6HB(10)=,F8.5)
 46    FORMAT(1X,6H +/-  ,E17.10,1X,
     *6HB( 6)=,F8.5,1X,6HB( 7)=,F8.5,1X,6HB( 8)=,F8.5,1X,
     *6HB( 9)=,F8.5,1X,
     *6HB(10)=,F8.5)
410     FORMAT(25X,2HB(,I2,2H)=,F8.5,1X,
     *2HB(,I2,2H)=F8.5,1X,
     *2HB(,I2,2H)=F8.5,1X,
     *2HB(,I2,2H)=F8.5,1X,
     *2HB(,I2,2H)=F8.5)
431     FORMAT(25X,2HB(,I2,2H)=,F8.5)
432     FORMAT(25X,2HB(,I2,2H)=,F8.5,1X,
     *2HB(,I2,2H)=F8.5)
433     FORMAT(25X,2HB(,I2,2H)=,F8.5,1X,
     *2HB(,I2,2H)=F8.5,1X,
     *2HB(,I2,2H)=F8.5)
434     FORMAT(25X,2HB(,I2,2H)=,F8.5,1X,
```

```
     *2HB(,I2,2H)=F8.5,1X,
     *2HB(,I2,2H)=F8.5,1X,
     *2HB(,I2,2H)=F8.5)
      END
```

```
      FUNCTION ANORZF(N,Z)
C
      COMMON/WENORM/WEN(100)
      COMMON/X2/X2(100)
      DIMENSION Z(N,N)
C
      DO 27 J=1,N
      W4=0.
      DO 26 I=1,N
   26 W4=W4+ABSF(Z(J,I))/WEN(I)
   27 X2(J)=W4*WEN(J)
      ANORZF=AMAXF(N,X2)
      RETURN
      END
```

```
      FUNCTION AMAXF(M,X5)
C
      DIMENSION X5(M)
C
      AMAX=0.
      DO 1 I=1,M
      AB=ABSF(X5(I))
      IF(AMAX-AB) 2,2,1
    2 AMAX=AB
    1 CONTINUE
      AMAXF=AMAX
      RETURN
      END
```

```
      FUNCTION SQRTF(X)
      SQRTF=SQRT(X)
      RETURN
      ENTRY ABSF
      IF(X) 1,2,2
    1 SQRTF=-X
      RETURN
    2 SQRTF=X
      RETURN
      ENTRY EXPF
      SQRTF=EXP(X)
      RETURN
      END
```


ПРИЛОЖЕНИЕ 2. ПРОГРАММА EXAMP

```
      PROGRAM EXAMP(INPUT,OUTPUT)
      DIMENSION Z(20,20),ZL(20,20),YR(100),XX(20)
     COMMON/KSEL/KPI,KPF,KGS,L/DROPX/DX(100)
C
C              EXAMPLE  1
C
C           SOLVING OF THE SYSTEM
C                X+Y**2=0    (1)
C                X**2+Y=2    (2)
C   BY MEANS OF   A R P  WITH INITIAL APPROXIMATIONS
C                X=-0.5 AND Y=-0.5
C
      NP=1
      M=2
      N=2
      NN=N
      K=4
      KK=1
       IT=-30
       E=1.
       T=0.00001
        A1=1.
       A2=1
       KPI=1
       KP=1
      YR(1)=2.
      YR(2)=0.
      XX(1)=-0.5
      XX(2)=-0.5
       PRINT 101
        CALL  REGN
     *(NP,M,N,NN,K,KK,IT,KP,T,AD,S,TT,D,E,Z,ZL,YR,XX,A1,A2)
C
C              EXAMPLE  2
C
C
C
       IT=30
       KGS=1
       KPF=0
       T=-0.0000000001
       AD=1.
       S=.1
        TT=2.
       D=0.000000001
        XX(1)=-0.5
       XX(2)=-0.4
       PRINT 102
        CALL  REGN
     *(NP,M,N,NN,K,KK,IT,KP,T,AD,S,TT,D,E,Z,ZL,YR,XX,A1,A2)
C
C              EXAMPLE  3
C
C     EXAMPLE OF  C.  L A N C Z O S
C
```

```
C
      T=0.000001
      NP=2
      M=72
      N=8
      NN=6
      K=3
      KK=3
      IT=-30
      S=0.
      AD=0.
      D=0.
      E=10.
      KPF=1
      YR(1)=2.5134$YR(2)=1.$YR(3)=0.0
      YR(4)=2.044333373$YR(5)=1.$YR(6)=.05
      YR(7)=1.668404436$YR(8)=1.$YR(9)=.1
      YR(10)=1.366418021$YR(11)=1.$YR(12)=.15
      YR(13)=1.123232487$YR(14)=1.$YR(15)=.2
      YR(16)=0.926889718$YR(17)=1.$YR(18)=.25
      YR(19)=0.767933856$YR(20)=1.$YR(21)=.3
      YR(22)=0.638877552$YR(23)=1.$YR(24)=.35
      YR(25)=0.533783531$YR(26)=1.$YR(27)=.4
      YR(28)=0.447936361$YR(29)=1.$YR(30)=.45
      YR(31)=0.377584788$YR(32)=1.$YR(33)=.5
      YR(34)=0.319739319$YR(35)=1.$YR(36)=.55
      YR(37)=0.272013077$YR(38)=1.$YR(39)=.6
      YR(40)=0.232496552$YR(41)=1.$YR(42)=.65
      YR(43)=0.199658954$YR(44)=1.$YR(45)=.7
      YR(46)=0.172270412$YR(47)=1.$YR(48)=.75
      YR(49)=0.149340566$YR(50)=1.$YR(51)=.8
      YR(52)=0.130070020$YR(53)=1.$YR(54)=.85
      YR(55)=0.113811932$YR(56)=1.$YR(57)=.9
      YR(58)=0.100041558$YR(59)=1.$YR(60)=.95
      YR(61)=0.088332090$YR(62)=1.$YR(63)=1.
      YR(64)=0.078335440$YR(65)=1.$YR(66)=1.05
      YR(67)=0.069766937$YR(68)=1.$YR(69)=1.1
      YR(70)=0.062393125$YR(71)=1.$YR(72)=1.15
      XX(1)=0.12
      XX(2)=1.1
      XX(3)=0.9
      XX(4)=0.6
      XX(5)=1.3
      XX(6)=2.8
      XX(7)=4.7
      XX(8)=4.7
      KGS=0
      KP=0
      PRINT 103
      CALL   REGN
     *(NP,M,N,NN,K,KK,IT,KP,T,AD,S,TT,D,E,Z,ZL,YR,XX,A1,A2)
      XX(4)=0.
      DX(4)=1.
      DX(8)=1.
      E=0.05
```



```
      KPF=3
      L=3
      CALL   REGN
     *(NP,M,N,NN,K,KK,IT,KP,T,AD,S,TT,D,E,Z,ZL,YR,XX,A1,A2)
C
C
 101  FORMAT(///20X,22HSOLUTION OF EXAMPLE  1,/)
 102  FORMAT(///20X,22HSOLUTION OF EXAMPLE  2,/)
 103  FORMAT(///20X,22HSOLUTION OF EXAMPLE  3,/)
      END
```

```
      SUBROUTINE RELADI (N,NP,INDEX,A,Y,DF)
      DIMENSION A(N),DF(N)
      COMMON/X/X(10)
      GO TO (100,200),NP
100   CONTINUE
      GO TO (1,2),INDEX
1     Y=A(1)**2+A(2)
      DF(1)=2.*A(1)
      DF(2)=1.
      RETURN
2     Y=A(1)+A(2)**2
      DF(1)=1.
      DF(2)=2.*A(2)
      RETURN
200   CONTINUE
      Y=A(1)*EXPF(-A(5)*X(1))+A(2)*EXPF(-A(6)*X(1))+
     *A(3)*EXPF(-A(7)*X(1))+A(4)*EXPF(-A(8)*X(1))
      DF(1)=EXPF(-A(5)*X(1))
      DF(2)=EXPF(-A(6)*X(1))
      DF(3)=EXPF(-A(7)*X(1))
      DF(4)=EXPF(-A(8)*X(1))
      DF(5)=-A(1)*X(1)*DF(1)
      DF(6)=-A(2)*X(1)*DF(2)
      DF(7)=-A(3)*X(1)*DF(3)
      DF(8)=-A(4)*X(1)*DF(4)
      RETURN
      END
```

```
                SOLUTION OF EXAMPLE  1

EXTITT INTITT      RO          MAX DEFECT       HI SQ           TAU            COND          EPS
   0      0     .2000000E+01  .2250000E+01  .5125000E+01  .4000000E+01 0.                .1000000E+01
      UNKNOWNS
X( 1)= -.5000000000E+00 X( 2)= -.5000000000E+00
EXTITT INTITT      RO          MAX DEFECT       HI SQ           TAU            COND          EPS
   1      0     .1432000E+01  .1290000E+01  .2456200E+01  .6240000E+01  .5000000E+01  .1000000E+01
      UNKNOWNS
X( 1)= -.9000000000E+00 X( 2)= -.1000000000E+00
EXTITT INTITT      RO          MAX DEFECT       HI SQ           TAU            COND          EPS
   2      0     .1233396E+01  .5777597E+00  .6318340E+00  .4899042E+01  .1315900E+02  .5288902E+00
      UNKNOWNS
X( 1)= -.9065053713E+00 X( 2)=  .6004882992E+00
EXTITT INTITT      RO          MAX DEFECT       HI SQ           TAU            COND          EPS
   3      0     .1927782E+00. .7944503E-01  .7937457E-02  .5080119E+01  .2100233E+01  .5643872E+00
      UNKNOWNS
X( 1)= -.9595013318E+00 X( 2)=  .9999121671E+00
EXTITT INTITT      RO          MAX DEFECT       HI SQ           TAU            COND          EPS
   4      0     .1593085E-02  .7626278E-03  .5861363E-06  .5003436E+01  .1100731E+01  .9316055E-01
      UNKNOWNS
X( 1)= -.9998744257E+00 X( 2)=  .1000318476E+01
EXTITT INTITT      RO          MAX DEFECT       HI SQ           TAU            COND          EPS
   5      0     .4718069E-06  .2226742E-06  .5028384E-13  .5000001E+01  .1001066E+01  .7958687E-03
      UNKNOWNS
X( 1)= -.9999999660E+00 X( 2)=  .1000000094E+01
EXTITT INTITT      RO          MAX DEFECT       HI SQ           TAU            COND          EPS
   6      0     .2842171E-13  .1421085E-13  .2019484E-27  .5000000E+01  .1000000E+01  .2359034E-06
      UNKNOWNS
X( 1)= -.1000000000E+01 X( 2)=  .1000000000E+01
```



```
                SOLUTION OF EXAMPLE  2

EXTITT INTITT      RO          MAX DEFECT       HI SQ           TAU            COND          EPS
   0      0     .1877997E+01  .2150000E+01  .4738100E+01  .3799997E+01 0.                .1000000E+01
      UNKNOWNS
X( 1)= -.5000000000E+00 X( 2)= -.4000000000E+00
EXTITT INTITT      RO          MAX DEFECT       HI SQ           TAU            COND          EPS
   1     26     .1877997E+01  .1106721E+01  .1254023E+01  .3799997E+01  .2166257E+02  .1810000E+00
      UNKNOWNS
X( 1)= -.3846511819E+00 X( 2)=  .7453220392E+00
EXTITT INTITT      RO          MAX DEFECT       HI SQ           TAU            COND          EPS
   2     21     .1022257E+01  .1994843E+00  .3979786E-01  .3943339E+01  .2626411E+01  .6010000E+00
      UNKNOWNS
X( 1)= -.9598504091E+00 X( 2)=  .1076724081E+01
EXTITT INTITT      RO          MAX DEFECT       HI SQ           TAU            COND          EPS
   3      2     .4276163E+00  .5540618E-02  .3316590E-04  .5871090E+01  .1307703E+01  .1000000E-02
      UNKNOWNS
X( 1)= -.9995213883E+00 X( 2)=  .1002527808E+01
EXTITT INTITT      RO          MAX DEFECT       HI SQ           TAU            COND          EPS
   4      2     .1268002E-01  .7468449E-05  .5604208E-10  .5026243E+01  .1007227E+01  .1000000E-02
      UNKNOWNS
X( 1)= -.9999987120E+00 X( 2)=  .1000003090E+01
EXTITT INTITT      RO          MAX DEFECT       HI SQ           TAU            COND          EPS
   5      2     .1545105E-04  .1484338E-08  .2208409E-17  .5000014E+01  .1000012E+01  .1000000E-02
      UNKNOWNS
X( 1)= -.9999999997E+00 X( 2)=  .1000000001E+01
EXTITT INTITT      RO          MAX DEFECT       HI SQ           TAU            COND          EPS
   6      2     .3040416E-08  .2771117E-12  .7759867E-25  .5000014E+01  .1000015E+01  .1000000E-02
      UNKNOWNS
X( 1)= -.1000000000E+01 X( 2)=  .1000000000E+01
```

SOLUTION OF EXAMPLE 3

I

```
EXTITT INTITT       RO         MAX DEFECT       HI SQ           TAU            COND          EPS
   0      0    .1610487E+01  .2102651E+00  .3433033E+00  .2325818E+02 0.                .1000000E+02
      UNKNOWNS
X( 1)=  .1200000000E+00 X( 2)=  .1100000000E+01 X( 3)=  .9000000000E+00 X( 4)=  .6000000000E+00
X( 5)=  .1300000000E+01 X( 6)=  .2800000000E+01 X( 7)=  .4700000000E+01 X( 8)=  .4700000000E+01
EXTITT INTITT       RO         MAX DEFECT       HI SQ           TAU            COND          EPS
  30      0    .2942844E-03  .3836611E-04  .1616880E-07  .2483070E+02  .2857858E+08  .1392668E-05
      UNKNOWNS
X( 1)=  .9690149112E-01 X( 2)=  .8524293717E+00 X( 3)=  .9411003101E+00 X( 4)=  .6229617757E+00
X( 5)=  .1012465292E+01 X( 6)=  .2999541247E+01 X( 7)=  .5051622849E+01 X( 8)=  .4905244549E+01
```

II

```
EXTITT INTITT       RO         MAX DEFECT       HI SQ           TAU            COND          EPS
   0      0    .2430524E+01  .6229688E+00  .1001050E+01  .2448020E+02 0.                .5000000E-01
      UNKNOWNS
X( 1)=  .9690149112E-01 X( 2)=  .8524293717E+00 X( 3)=  .9411003101E+00 X( 4)= 0.
X( 5)=  .1012465292E+01 X( 6)=  .2999541247E+01 X( 7)=  .5051622849E+01 X( 8)=  .4905244549E+01
EXTITT INTITT       RO         MAX DEFECT       HI SQ           TAU            COND          EPS
  17      L    .2700809E-13  .5807772E-09  .1173330E-17  .2498246E+02  .9635190E+09  .3550221E-12
      UNKNOWNS
X( 1)=  .9510015985E-01 B( 1)= 1.00000 B( 2)=  .97587 B( 3)= -.98753 B( 4)= 0.00000 B( 5)=  .99948
 +/-    .1527730088E-06 B( 6)=  .99560 B( 7)=  .97924 B( 8)= 0.00000
X( 2)=  .8607004929E+00 B( 1)=  .97587 B( 2)= 1.00000 B( 3)= -.99807 B( 4)= 0.00000 B( 5)=  .96864
 +/-    .3887842592E-06 B( 6)=  .99200 B( 7)=  .99976 B( 8)= 0.00000
X( 3)=  .1557599347E+01 B( 1)= -.98753 B( 2)= -.99807 B( 3)= 1.00000 B( 4)= 0.00000 B( 5)= -.98217
 +/-    .5388349604E-06 B( 6)= -.99791 B( 7)= -.99885 B( 8)= 0.00000
X( 4)= 0.                         IS DROPPED UNKNOWN
 +/-   0.
X( 5)=  .1000000804E+01 B( 1)=  .99948 B( 2)=  .96864 B( 3)= -.98217 B( 4)= 0.00000 B( 5)= 1.00000
 +/-    .7867561394E-06 B( 6)=  .99215 B( 7)=  .97261 B( 8)= 0.00000
X( 6)=  .3000001100E+01 B( 1)=  .99560 B( 2)=  .99200 B( 3)= -.99791 B( 4)= 0.00000 B( 5)=  .99215
 +/-    .9538630806E-06 B( 6)= 1.00000 B( 7)=  .99376 B( 8)= 0.00000
X( 7)=  .5000000393E+01 B( 1)=  .97924 B( 2)=  .99976 B( 3)= -.99885 B( 4)= 0.00000 B( 5)=  .97261
 +/-    .3166588605E-06 B( 6)=  .99376 B( 7)= 1.00000 B( 8)= 0.00000
X( 8)=  .4905244549E+01           IS DROPPED UNKNOWN
 +/-   0.
```