



Белов А.А.

Программы SuFaReC для  
сверхбыстрого расчета  
эллиптических уравнений в  
прямоугольной области

**Рекомендуемая форма библиографической ссылки:** Белов А.А. Программы SuFaReC для сверхбыстрого расчета эллиптических уравнений в прямоугольной области // Препринты ИПМ им. М.В.Келдыша. 2015. № 44. 12 с. URL: <http://library.keldysh.ru/preprint.asp?id=2015-44>

РОССИЙСКАЯ АКАДЕМИЯ НАУК  
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ  
ИМ. М.В. КЕЛДЫША

А. А. Белов

ПРОГРАММЫ SUFAREC  
ДЛЯ СВЕРХБЫСТРОГО РАСЧЕТА  
ЭЛЛИПТИЧЕСКИХ УРАВНЕНИЙ  
В ПРЯМОУГОЛЬНОЙ ОБЛАСТИ

Москва, 2015

УДК 519.6

А. А. Белов. Программы SuFaReC для сверхбыстрого расчета эллиптических уравнений в прямоугольной области.

Предложена программная реализация сверхбыстрого метода решения с гарантированной точностью для эллиптических задач в двумерной и трехмерной прямоугольных областях. В качестве среды реализации выбран пакет Matlab. Дано описание особенностей алгоритма и представлены коды программ. Работа этого алгоритма иллюстрируется на примере задач с узким пограничным слоем.

*Ключевые слова:* эллиптические уравнения, счет на установление, логарифмический набор шагов, эволюционная факторизация, оценки погрешности, метод Ричардсона, пакет программ, пограничный слой.

A. A. Belov. SuFaReC programs for superfast elliptic problems solving in rectangular domain.

A program realization is proposed for superfast elliptic problems solving with a posteriori error estimation in two-dimensional and three-dimensional rectangular domains. The programs are implemented in MatLab environment. The algorithm features are described in detail and program codes are presented. The algorithm performance is illustrated for thin boundary layer problems.

*Keywords:* elliptic problems, relaxation count, logarithmic set of steps, evolutionary factorization, error estimation, Richardson method, program complex, boundary layer.

Работа поддержана грантом РФФИ №14-01-00161.

# 1 Введение

Рассмотрим задачу Дирихле для эллиптического уравнения

$$\begin{aligned}\mu^2 Lu - \varkappa u &= -f(\mathbf{r}), \quad \mathbf{r} \in G, \\ u(\mathbf{r}) &= \varphi(\mathbf{r}), \quad \mathbf{r} \in \partial G.\end{aligned}\tag{1}$$

Мы предполагаем, что оператор

$$Lu = \sum_{\alpha} L_{\alpha} u, \quad L_{\alpha} u = \frac{\partial}{\partial x_{\alpha}} \left( k_{\alpha}(\mathbf{r}) \frac{\partial u}{\partial x_{\alpha}} \right)\tag{2}$$

не содержит смешанных производных, а область  $G$  является прямоугольной. При этом коэффициенты  $k_{\alpha}$  будем считать переменными, так что данная постановка достаточно содержательна. Если  $\varkappa = 0$ , то получаем уравнение Пуассона, если  $\varkappa \neq 0$ , то уравнение Гельмгольца. Если при этом  $\mu \ll 1$ , то задача будет сингулярно возмущенной. Последний случай требует специального рассмотрения.

Задача (1)–(2) решается разностными методами. Сетки выберем прямоугольными, но неравномерными и запишем следующую консервативную разностную схему [1]

$$\mu^2 \sum_{\alpha} \Lambda_{\alpha} u - \varkappa u = -f,\tag{3}$$

где трехточечный оператор  $\Lambda_x$  имеет вид

$$(\Lambda_x u)_n = \frac{2}{h_{x,n+1/2} + h_{x,n-1/2}} \left[ \frac{k_{x,n+1/2}}{h_{x,n+1/2}} (u_{n+1} - u_n) - \frac{k_{x,n-1/2}}{h_{x,n-1/2}} (u_n - u_{n-1}) \right],\tag{4}$$

в (4) оставлен только индекс по координате  $x$ . Выражение для  $\Lambda_y$  (и  $\Lambda_z$  в трехмерном случае) записывается аналогично.

Для решения разностного уравнения (3)–(4) разработан эффективный алгоритм [2]–[6]. Решение находится с учетом установившегося по экономической эволюционно-факторизованной схеме со специальным линейно-тригонометрическим набором шагов в логарифмической шкале. Данный алгоритм требует значительно меньшего объема вычислений, чем общеизвестные методы, что позволяет строить решение с высокой точностью при умеренном расходе машинного времени. Одновременно с разностным решением вычисляется апостериорная асимптотически точная оценка погрешности итераций. Такие расчеты проводятся на последовательности сгущающихся вдвое сеток. Применение к ним метода Ричардсона позволяет получить решение задачи (1)–(2) и одновременно дать апостериорную асимптотически точную оценку его погрешности.

В настоящей работе представлена реализация данного алгоритма в виде пакета программ в среде Matlab. Пакет может быть востребован в силу большой актуальности задач для эллиптических уравнений описанного

типа и широкой распространенности используемого программного обеспечения. Поскольку код не содержит специализированных функций, присутствующих только в среде Matlab, то он допускает несложный перенос на более эффективные языки программирования. Эта процедура сводится только к замене синтаксиса. Ниже мы опишем наиболее важные идеи, лежащие в основе данного пакета.

## 2 Эволюционная факторизация

Будем решать задачу (3)–(4) с учетом установления. Это значит, что нужно найти стационарный предел решения нестационарной разностной схемы

$$\frac{\hat{u} - u}{\tau} = \mu^2 \sum_{\alpha} \Lambda_{\alpha} \frac{\hat{u} + u}{2} - \varkappa u + f. \quad (5)$$

где  $\hat{u}$  – решение на новом временном слое. Для экономичного решения последней нужно применять тот или иной способ факторизации. Наиболее удобной для практики оказывается *эволюционная факторизация* [2]

$$\prod_{\alpha} \left[ E - \frac{\tau}{2} (\mu^2 \Lambda_{\alpha} - \varkappa_{\alpha}) \right] \frac{\hat{u} - u}{\tau} = \sum_{\alpha} (\mu^2 \Lambda_{\alpha} - \varkappa_{\alpha}) u + f, \quad \sum_{\alpha} \varkappa_{\alpha} = \varkappa. \quad (6)$$

Эта схема имеет аппроксимацию  $O(\tau^2 + \sum h^2)$ , является безусловно устойчивой и условно асимптотически устойчивой. К числу ее достоинств относится и то, что она единообразно записывается для произвольного числа измерений. Поскольку операторы  $\Lambda_{\alpha}$  являются трехдиагональными, решение схемы (6) сводится к последовательности 2 одномерных прогонок в двумерном случае и 3 – в трехмерном.

Проанализируем трудоемкость одного шага по времени и сравним ее с трудоемкостью одной итерации метода сопряженных градиентов, который является наилучшим из общих итерационных методов. Одна одномерная прогонка требует 9 операций на один узел сетки: 4 умножения, 4 сложения и 1 деление. Одна итерация метода сопряженных градиентов содержит 3 скалярных произведения ( $3N$  умножений и  $3N$  сложений), 3 вычисления векторов по известным формулам (еще  $3N$  умножений и  $3N$  сложений) и 1 умножение матрицы на вектор. Если в строке матрицы в среднем  $k$  ненулевых элементов, то это дает еще  $kN$  умножений и  $kN$  сложений. Поэтому на один узел сетки приходится  $k + 6$  операций. Для двумерных задач  $k = 5$ , для трехмерных –  $k = 7$ .

Для большей наглядности сравнения возьмем “среднее” между двумерным и трехмерным случаями. Тогда на одну итерацию метода сопряженных градиентов потребуется по 12 умножений и сложений, а на 1 шаг схемы (6) – 2.5 деления и по 10 умножений и сложений. Сравнение времен,

за которые выполняются операции сложения, умножения и деления, показывает, что один шаг схемы (6) требует такого же программного времени, что и одна итерация метода сопряженных градиентов. Поскольку в данном методе число итераций оказывается в  $\sim 100$  раз меньше, чем в методе сопряженных градиентов, он оказывается несоизмеримо более быстрым.

### 3 Логарифмические наборы шагов

Эти наборы эвристически обобщают строгий результат о постоянном оптимальном шаге для счета на установление [3] – [6]. Набор шагов с производящей функцией  $f(s)$  предлагается строить в виде

$$\ln \tau_s = \frac{1}{2} \ln (\tau_{\max} \tau_{\min}) + \frac{1}{2} \ln \frac{\tau_{\max}}{\tau_{\min}} f(s), \quad 0 \leq s \leq S. \quad (7)$$

Граничные шаги  $\tau_{\max}$  и  $\tau_{\min}$  вычисляются через границы спектра по направлениям, для которых построены эффективные оценки. Например, в двумерном случае  $\tau_{\max} = 2 / \min\{\lambda_x, \lambda_y\}$ ,  $\tau_{\min} = 2 / \max\{\lambda_x, \lambda_y\}$ . В трехмерном случае формулы выглядят сложнее.

В работах [4] – [6] предложен *линейно-тригонометрический набор*, являющийся практически оптимальным вариантом логарифмического набора

$$f_{\text{ЛТ}}(s) = \frac{\pi}{\pi + 2} \left( \frac{2s}{S} - 1 \right) - \frac{2}{\pi + 2} \cos \left( \frac{\pi s}{S} \right). \quad (8)$$

Для набора (7)–(8) построена следующая априорная оценка сходимости итераций:

$$S = \frac{4}{\pi^2 + 2\pi} \ln \frac{\lambda_{\max}}{\lambda_{\min}} \ln \frac{1}{\varepsilon} \approx 13 \ln N, \quad (9)$$

где  $S$  – число итераций, достаточное для достижения точности  $\varepsilon$  на сетке с  $N$  шагами по каждому направлению. Оценка (9) означает несколько десятков (но не более 100) итераций, что существенно превосходит по дешевизне все прямые и итерационные методы общего вида. Например, метод сопряженных градиентов требует  $S \approx 10N$ , что на практике приводит к тысячам и десяткам тысяч итераций (при том, что трудоемкость одной итерации такова же). Сопоставимую с (9) скорость имеют лишь быстрое преобразование Фурье и нечетно-четная редукция, но они применимы лишь к очень узкому классу задач ( $k = \text{const}$ ,  $h = \text{const}$ ,  $N = 2^m$ ).

Экспоненциальный характер сходимости итераций позволяет построить *апостериорные оценки точности*. Пусть  $\{\tau^q\}$  – последовательность двукратно сгущающихся логарифмических сеток, индекс  $q$  соответствует номеру сетки (начиная с самой редкой  $q = 1$ ). Обозначим через  $\{U^q\}$  соответствующую последовательность решений на этих сетках. Тогда имеют

место следующие оценки:

$$\|U^q - u\| \approx \|U^{q+1} - U^q\|, \quad (10)$$

$$\|U^{q+1} - u\| \approx \|U^{q+1} - U^q\|^3 / \|U^q - U^{q-1}\|^2. \quad (11)$$

Оценка (10) называется интерполяционной, оценка (11) – экстраполяционной. Обе эти оценки являются асимптотически точными и работают до выхода на фон ошибок округления. На практике априорная оценка (9) используется для построения последовательности логарифмических сеток, а апостериорные оценки (10)–(11) – для нахождения фактической точности.

Рекомендуется следующая процедура. Для априорной оценки  $S$  необходимо задать требуемую точность  $\varepsilon$ . Здесь возможны две ситуации. 1) Точность  $\varepsilon_{\text{п}}$ , не превосходящая фоновую  $\varepsilon_{\text{ф}}$ , может быть задана пользователем. 2) Требуется рассчитать как можно точнее. В последнем случае необходимо оценить фон ошибок округления. Для этого существуют мажорантные оценки. Фон получается умножением ошибки единичного округления на число обусловленности матрицы. В наших расчетах наиболее просто вычисляется несколько завышенное спектральное число обусловленности  $\zeta_{\lambda} = \sum \lambda_{\alpha, \text{max}} / \sum \lambda_{\alpha, \text{min}}$ . Тогда  $\varepsilon_{\text{ф}} = 10^{-16.2} \zeta_{\lambda}$  (для 64-разрядного программного обеспечения).

Положим  $\varepsilon = \max \{\varepsilon_{\text{п}}, \varepsilon_{\text{ф}}\}$ . Исходя из этой точности и оценок границ спектра, вычислим требуемое  $S$ . Будем делить это число рекуррентно на 2 до тех пор, пока не получится число, немного меньшее небольшого целого  $S_0$  (1...5). Полученное  $S_0$  следует взять в качестве начального и применять процедуру сгущения.

Процедура сгущения применяется до достижения требуемого  $S$ . При этом все погрешности, кроме последней, вычисляются по оценке (10). Эта оценка надежна, поскольку применяется вдали от фона. Для последнего вычисления применим экстраполяцию (11) и сравним полученную оценку  $\varepsilon_{\text{э}}$  с фоновой  $\varepsilon_{\text{ф}}$ . В качестве погрешности последнего вычисления выберем величину  $\max \{\varepsilon_{\text{э}}, \varepsilon_{\text{ф}}\}$ . Если же логарифмических сеток всего 2 и оценку  $\varepsilon_{\text{э}}$  построить не удастся, то точность решения на второй сетке заведомо не хуже, чем та, по которой вычислялось  $S$ . Для визуального контроля удобно выводить оценки погрешностей на график.

Отметим еще одну особенность оценок (10)–(11): они справедливы только для норм, поточечно их использовать нельзя. В предлагаемых программах пользователь может выбрать один из трех вариантов различных точных норм: норму  $C$ ,  $L_2$  или среднеквадратичную. Формально обе последние нормы можно трактовать как среднеквадратичные, однако норма  $L_2$  вычисляется как среднее с весами, равными шагам сетки, а в истинно среднеквадратичной норме веса являются единичными. При этом первая более физична, а вторая более тонко отслеживает поведение решения в тех областях, где мы вынуждены сгущать сетку.

## 4 Сгущение сеток по пространству

Проведем расчеты на последовательности сгущающихся сеток по пространству с числами узлов  $N, 2N, 4N \dots$  по каждому направлению, начиная с некоторого небольшого  $N$ . Поточечно сравнивая решения на этих сетках, можно получить апостериорную асимптотически точную оценку погрешности сеточных решений, пользуясь *методом Ричардсона* [7].

Суть последнего сводится к следующему. Пусть  $R_N$  и  $R_{2N}$  – погрешности решений  $u_N$  и  $u_{2N}$ , вычисленных на сетках с  $N$  и  $2N$  узлами соответственно. Пусть порядок точности схемы равен  $p$ , то есть  $R_N = u_N - u \sim h^p$ . Тогда справедлива оценка

$$R_{2N} = (u_N - u_{2N}) / (2^p - 1). \quad (12)$$

Эта оценка является поточечной и относится к сетке с числом узлов  $2N$ .

Одновременно по скорости убывания погрешности, вычисленной из (12), можно подтверждать порядок фактической точности. Для этого достаточно построить график нормы погрешности как функции числа узлов по одному направлению в двойном логарифмическом масштабе. Из степенного закона сходимости следует, что  $\lg \|R_N\| = const + p \lg N$ , то есть этот график должен асимптотически выходить на прямую с наклоном  $p$ . Это справедливо и для нецелых  $p$ , что соответствует ограниченной гладкости решения. Такой подход позволяет обойтись без использования априорных оценок точности, которые зачастую сложно построить.

Контролировать процесс выхода на асимптотический режим удобнее всего визуально по наклону графика. Этот процесс можно автоматизировать. Но на практике автоматизация требует большего числа сгущений сеток, чем визуальный контроль, а в многомерном случае каждое лишнее сгущение сильно увеличивает объем расчетов. Отметим также, что для применения сгущений сеток коэффициенты и правые части должны быть дважды непрерывно дифференцируемы.

## 5 Сингулярно возмущенные задачи

Пусть  $\varkappa \neq 0$ ,  $\mu \ll 1$ . Решение таких задач имеет 3 характерных зоны: 1) пограничный слой толщины  $\sim \mu$ , в котором решение меняется очень сильно, 2) регулярную вдали от границы и 3) переходную, расположенную между пограничным слоем и регулярной частью. Переходная зона характеризуется большой кривизной решения, что также сложно для расчета.

Чтобы детально передать все эти зоны решения, нужно, чтобы в каждую из них попало примерно одинаковое число узлов сетки. Требуемым



свойством обладает сетка вида

$$x(\xi) = A \operatorname{th}[C\xi(1 + \xi^2/3)], \quad \xi \in [-1, 1]. \quad (13)$$

Параметры  $A$  и  $C$  подбираются так, чтобы 1)  $x \in [-1, 1]$  и 2)  $x'(1) = \mu/(\mu + \varkappa)$ . Второе условие задает шаг на границе. Эти условия сводятся к одному трансцендентному уравнению, которое решается методом Ньютона.

В качестве окончательной сетки для задачи (3)–(4) выбирается декартово произведение сеток вида (13). Такая сетка подстраивается под особенности задачи и хорошо работает при любом соотношении  $\mu$  и  $\varkappa$ . Заметим также, что шаг сетки следует определять не как разность  $x_{n+1} - x_n$ , а через производную производящей функции в полуцелой точке  $h_{n+1/2} = x'(\xi_{n+1/2})/N$ . Такое определение можно трактовать как некоторую замену переменных. Оно обладает рядом преимуществ. В частности, точность оказывается несколько лучше. Кроме того, такое определение позволяет распространить метод на неограниченные области [5], [7].

## 6 Работа программ

Проиллюстрируем работу предлагаемых программ SuFaReC (SUperFAst RElaxation Count) на примере двумерного и трехмерного сингулярно возмущенного уравнения Гельмгольца.

### 6.1 Двумерная задача (программа SuFaReC\_2)

Положим в уравнении

$$\begin{aligned} \mu &= 10^{-2}, \quad k_x(x, y) = k_y(x, y) = 1, \quad \varkappa = 1, \\ f(x, y) &= \cos[\pi(x + y)^2/4] \cos[3\pi(y - x)/4]. \end{aligned} \quad (14)$$

Эти параметры задаются в программе переменными `mu`, `kx`, `ky`, `kapra`, `f` соответственно. Определим область  $G = [a_x, b_x] \times [a_y, b_y]$ ,  $a_x = a_y = -1$ ,  $b_x = b_y = 1$  (переменные `ax`, `ay`, `bx`, `by`) и зададим граничные условия (переменная `u_boundary`)

$$u(x, y) = 2.5(x + y), \quad (x, y) \in \partial G. \quad (15)$$

В качестве сеток по пространству выберем автоматически генерируемые сетки (13), задав флаг `grid_choice = 0`. Заметим, что пользователь может определить сетки самостоятельно. Для этого нужно выбрать `grid_choice = 1`, ввести производящие функции сеток `x_grid`, `y_grid` и их производные `x_diff`, `y_diff`.

Пусть самая редкая сетка содержит  $N_0 = 16$  узлов по  $x$  и  $K_0 = 16$  узлов по  $y$  (переменные `N0` и `K0` соответственно), и пусть расчет выполняется на

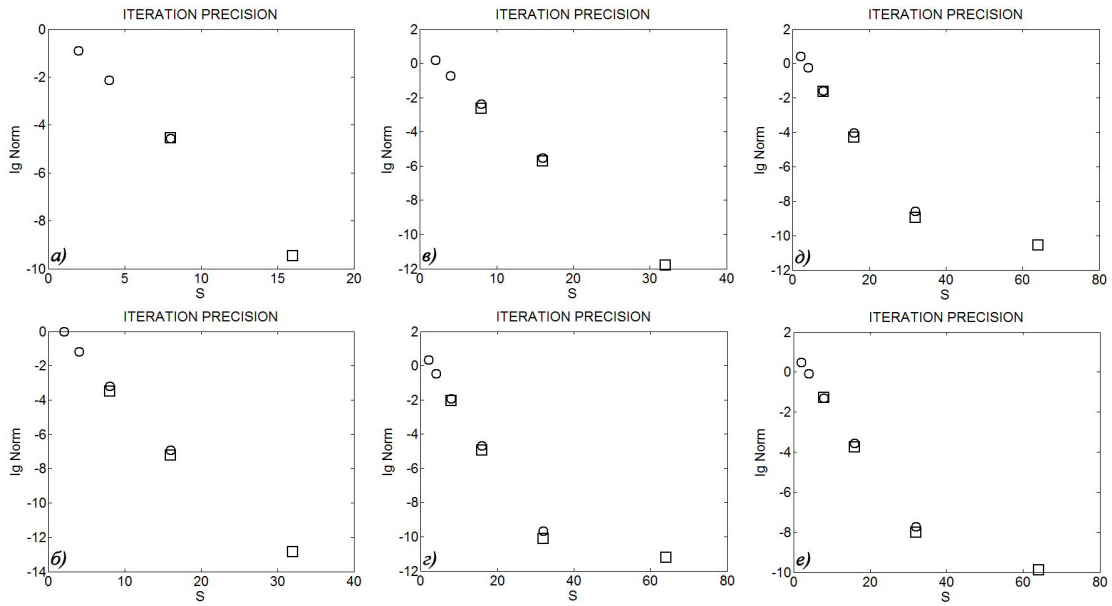


Рис. 1: Сходимость итераций в задаче (14)–(15). а)  $N = K = 16$ , б)  $N = K = 32$ , в)  $N = K = 64$ , г)  $N = K = 128$ , д)  $N = K = 256$ , е)  $N = K = 512$ ; круглые маркеры – оценка (10), квадратные – оценка (11).

$P = 6$  сетках (переменная  $P$ ), то есть производится 5 сгущений. Подчеркнем, что для вычисления оценки погрешности (12) нужно не менее 2 сеток, иначе программа выдаст сообщение об ошибке.

Зададим точность решения сеточных систем на каждой сетке  $\varepsilon_{\Pi} = 10^{-5}$  (переменная `epsilon_user`). В качестве нормы, в которой вычисляются все погрешности, выберем норму  $C$ , задав флаг `norm_choice = 0`. Если требуется норма  $L_2$ , то следует положить `norm_choice = 1`, если среднеквадратичная то `norm_choice = 2`.

В результате программа выводит следующие графики.

**1<sup>0</sup>** Графики сходимости итераций в полулогарифмическом масштабе, относящиеся к разным пространственным сеткам (рис. 1а–1е). Заметим, что в двумерном случае итерации сходятся быстрее, чем предсказывает оценка (9). Это связано с тем, что множитель роста двумерной гармоник в схеме (6) равен произведению 2 одномерных множителей роста, поэтому каждая гармоника подавляется сильнее, чем в одномерном случае. В трехмерном случае множитель роста имеет более сложную структуру, и такого ускорения сходимости не наблюдается.

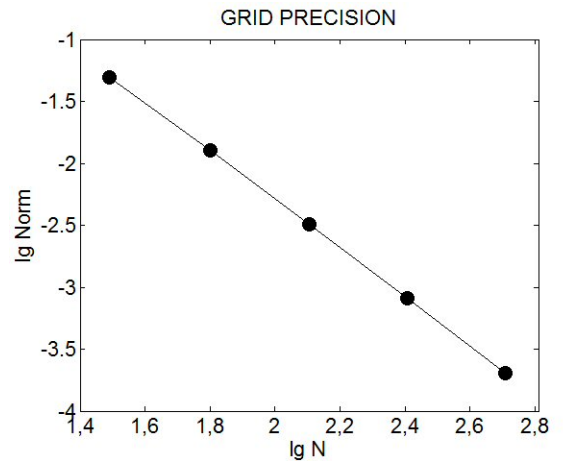


Рис. 2: Сходимость по пространству в задаче (14)–(15). Маркеры – оценка (12).

2<sup>0</sup> График сходимости при сгущении сеток по пространству (рис. 2), масштаб двойной логарифмический. Число точек на этом графике равно числу сгущений. Начало линии соответствует нерегулярной сходимости, но 3 последних точки ложатся на прямую с наклоном 2. Это подтверждает второй порядок точности схемы (3)–(4).

3<sup>0</sup> Общий вид решения в виде изолиний с фиксированным шагом (рис. 3) и сечение решения плоскостью, проходящей через ось  $u$  и диагональ области  $G$  (рис. 4). На двух последних графиках хорошо видны два угловых пограничных слоя.

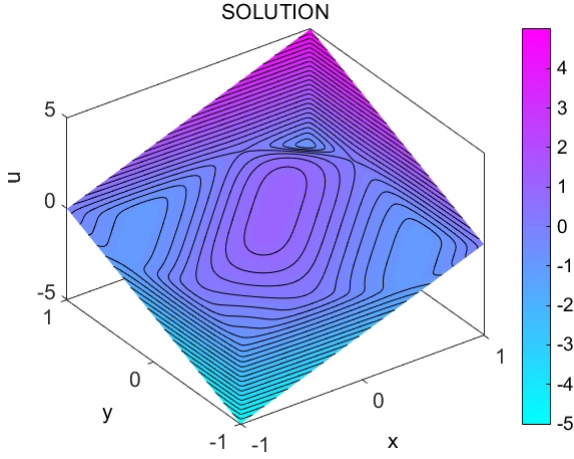


Рис. 3: Решение задачи (14) – (15).

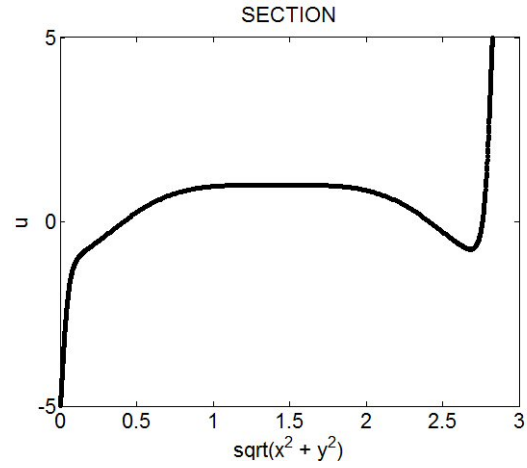


Рис. 4: Сечение решения задачи (14) – (15) плоскостью  $x = y$ .

## 6.2 Трехмерная задача (программа SuFaReC\_3)

Зададим параметры задачи

$$\begin{aligned}
 \mu &= 10^{-2}; \quad k_x(x, y, z) = k_y(x, y, z) = k_z(x, y, z) = 1, \quad \varkappa = 1; \\
 f(x, y, z) &= 1.5 \cos^2 [\pi(z + 1)(x^2 + y)] \cos [\pi(x + z)]; \\
 G &= [a_x, b_x] \times [a_y, b_y] \times [a_z, b_z]; \quad a_\alpha = -1, \quad b_\alpha = 1, \quad \alpha = x, y, z; \\
 u(x, y, z) &= 2.5(x + y + z), \quad (x, y, z) \in \partial G.
 \end{aligned} \tag{16}$$

Обозначения переменных аналогичны двумерной программе. Выберем автоматически генерируемые сетки (`grid_choice = 0`). Пусть самые редкие сетки по  $x, y, z$  содержат  $N_0 = K_0 = L_0 = 16$  узлов соответственно, и пусть расчет ведется на  $P = 4$  сетках. Зададим точность решения сеточных систем  $\varepsilon_{\Pi} = 10^{-5}$ , а для представления погрешностей выберем норму  $C$  (`norm_choice = 0`).

Результатом работы программы являются графики сходимости итераций (рис. 5а–5г), график сходимости при сгущении пространственных сеток (рис. 6), графики общего вида решения при фиксированном  $z =$

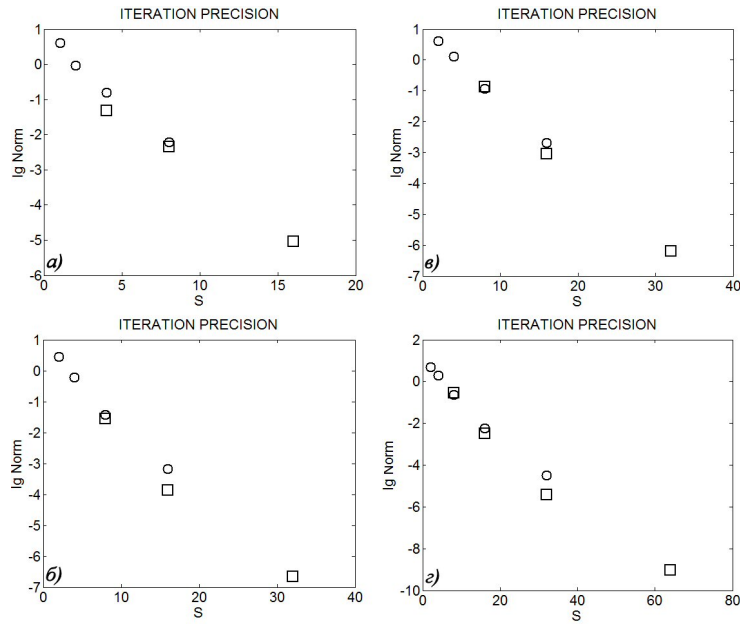


Рис. 5: Сходимость итераций в задаче (16). а)  $N = K = L = 16$ , б)  $N = K = L = 32$ , в)  $N = K = L = 64$ , г)  $N = K = L = 128$ ; круглые маркеры – оценка (10), квадратные – оценка (11).

$(a_z + b_z)/2$  (рис. 7) и сечения этого решения плоскостью, проходящей через ось  $u$  и диагональ квадрата на плоскости  $x, y$  (рис. 8). Из рис. 5 видно, что точности, достигнутые в итерационном процессе, близки к  $\varepsilon_{\text{п}}$ . Превышение связано со способом построения последовательности логарифмических сеток. На рис. 6 виден второй порядок точности схемы (3)–(4).

Листинги программ SuFaReC\_2 и SuFaReC\_3 расположены по ссылкам [http://keldysh.ru/attachment/2015-belov/SuFaReC\\_2.m](http://keldysh.ru/attachment/2015-belov/SuFaReC_2.m) и [http://keldysh.ru/attachment/2015-belov/SuFaReC\\_3.m](http://keldysh.ru/attachment/2015-belov/SuFaReC_3.m). Коды снабжены русскоязычными комментариями, поясняющими смысл обозначений. Пользователю необходимо ввести параметры задачи аналогично сделанному выше, дальнейший расчет происходит автоматически.

На выходе программы выдают графики, аналогичные рис. 1–8. Решения на сгущающихся пространственных сетках располагаются в массиве `solution`, первый индекс которого соответствует номеру сетки, начиная с самой редкой. Точности решения соответствующих сеточных систем записаны в массив `iteration_precision`, а точности ре-

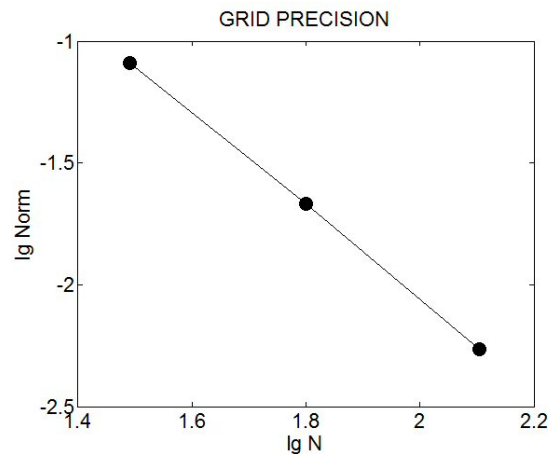


Рис. 6: Сходимость по пространству в задаче (16). Маркеры – оценка (12).

шения исходного дифференциального уравнения, определяемые методом Ричардсона, находятся в массиве `grid_precision`.

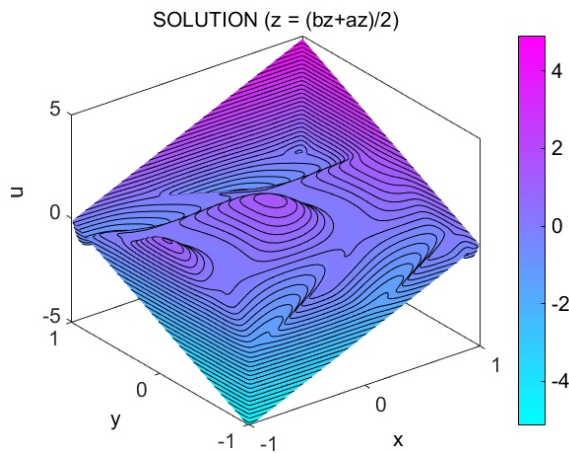


Рис. 7: Решение задачи (16) при фиксированном  $z = (a_z + b_z)/2$ .

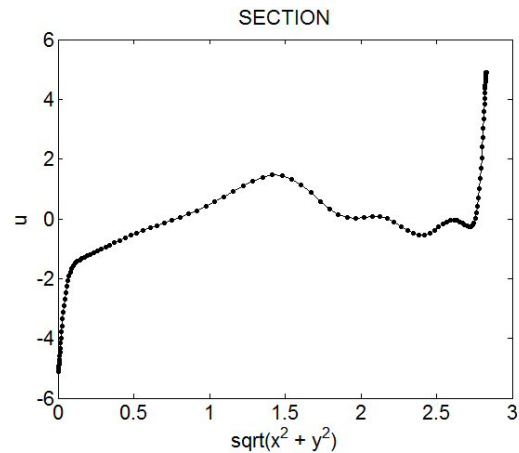


Рис. 8: Сечение решения задачи (16) при фиксированном  $z = (a_z + b_z)/2$  плоскостью  $x = y$ .

Автор выражает искреннюю благодарность Н. Н. Калиткину за ценные замечания и обсуждения.

## Список литературы

1. Самарский А. А., Гулин А. В. Численные методы, М.: Наука, 1989, 432 с.
2. Калиткин Н. Н. Улучшенная факторизация параболических схем // ДАН. 2005. Т. 402, №4. С. 467-471.
3. Болтнев А. А., Калиткин Н. Н., Качер О. А. Логарифмически сходящийся счет на установление // ДАН. 2005. Т. 404, №2. С. 177-180.
4. Белов А. А., Калиткин Н. Н. Эволюционная факторизация и сверхбыстрый счет на установление // Матем. моделир. 2014. Т. 26, №9. С. 47-64.
5. Калиткин Н. Н., Белов А. А. Аналог метода Ричардсона для логарифмически сходящегося счета на установление // ДАН. 2013. Т. 452, №3. С. 261-265.
6. Белов А. А., Калиткин Н. Н. Эволюционная факторизация и сверхбыстрый счет на установление // Препринты ИПМ им. М.В.Келдыша. 2013. №69. 36 с. URL: <http://library.keldysh.ru/preprint.asp?id=2013-69>
7. Калиткин Н. Н., Альшин А. Б., Альшина Е. А., Рогов Б. В. Вычисления на квазиравномерных сетках. М.: Физматлит, 2005, 224 с.