Всеволод Заостровский, 409 группа

Отчёт по задаче "Итерационные методы решения систем линейных уравнений".

1 Задача 1.

Для решения системы линейных уравнений:

$$
\begin{aligned}
&-\frac{y_{k+1}-2y_k+y_{k-1}}{h^2}+py_k=f_k, \qquad k=1,\dots,N-1;\\
&y_0=y_N=0;\\
&h=\frac{1}{N};\\
&p\geq 0.
\end{aligned}
$$

реализуйте метод Фурье (т.е. метод разложения по собственным векторам) для базисных функций:

$$\psi_k^{(n)}=\sin(\frac{\pi nk}{N}) \tag{1}$$

Решение.

Вычислим λ_n:

$A\psi^{(n)}{}_k=-\frac{\psi_{k+1}^{(n)}-2\psi_k^{(n)}+\psi_{k-1}^{(n)}}{h^2}+p\psi_k^{(n)}$

$=-\frac{\sin(\frac{\pi n(k+1)}{N})-2\sin(\frac{\pi nk}{N})+\sin(\frac{\pi n(k-1)}{N})}{h^2}+p\sin(\frac{\pi nk}{N})$

$=-\frac{\sin(\frac{\pi nk}{N})\cos(\frac{\pi n}{N})+\cos(\frac{\pi nk}{N})\sin(\frac{\pi n}{N})-2\sin(\frac{\pi nk}{N})+\sin(\frac{\pi nk}{N})\cos(\frac{\pi n}{N})-\cos(\frac{\pi nk}{N})\sin(\frac{\pi n}{N})}{h^2}+p\sin(\frac{\pi nk}{N})$

$=-\frac{\sin(\frac{\pi nk}{N})\cos(\frac{\pi n}{N})-2\sin(\frac{\pi nk}{N})+\sin(\frac{\pi nk}{N})\cos(\frac{\pi n}{N})}{h^2}+p\sin(\frac{\pi nk}{N})$

$=-\frac{2\sin(\frac{\pi nk}{N})(\cos(\frac{\pi n}{N})-1)}{h^2}+p\sin(\frac{\pi nk}{N})=\sin(\frac{\pi nk}{N})(-\frac{2(\cos(\frac{\pi n}{N})-1)}{h^2}+p).$

Отсюда:

$$\lambda_n=p-N^2 2(\cos(\frac{\pi n}{N})-1). \tag{2}$$

Кроме того, из указания к задаче:

$$c_n = \frac{\left(f, \psi^{(n)}\right)}{\lambda_n \left(\psi^{(n)}, \psi^{(n)}\right)} = \frac{2\left(f, \psi^{(n)}\right)}{\lambda_n}. \tag{3}$$

Решение же можно найти в виде:

$$y_k = \sum_{n=1}^{N-1} c_n \psi_k^{(n)}. \tag{4}$$

Огромные проблемы возникли всвязи с тем, что матрицей Фурье порядка N прадлагается называть матрицу, размерность которой $N+1$, при этом, подматрица из ненулевых элементов, для которой осмысленно ставить задачу решения СЛУ имеет размерность $N-1$. Об этом стоит помнить, во избежание изнурительной отладки.

Реализацию см. в файле "solution.h", незамысловатая система тестирования реализована в "test1.c".

2 Задача 2.

Для решения системы линейных уравнений из **Задачи 1.** реализуйте метод Ричардсона. Проведите тестирование программы, сравнив теоретическую и реальную скорость сходимости. Для этого на каждом k-ом шаге, $k = 0, 1, \ldots$ итерационного процесса сохраните в некоторый файл очередную тройку.

Решение. Реализацию см. в файле "solution.h", незамысловатая система тестирования реализована в "test2.c".

Фактически реализован функционал (правда, не самый эффективный), позволяющий применить метод к любой матрице. Очень остро здесь стоит вопрос определения m и M (см. файл "../LinAlg.pdf"). Для матрицы из задачи 1 есть явная формула собственных значений, благодаря чему удалось добиться хорошего приближения к теоретической сходимости. Однако, если бы не было известной формы, то найти их с приемлемой точностью представлялось бы задачей, вычислительная сложность которой сравнима со сложностью решения всей системы. При этом, неправильное tau, как показали многочисленные отладочные тесты, не включенные в отчет, может очень сильно испортить сходимость.

По результатам тестирования был построен график (для вывода ./LS2 50 1000 100 out.txt, те матрица порядка 50, до 1000 итераций с $p = 100$). См. рисунок.

3 Задача 3.

Для решения системы линейных уравнений:

$$
\begin{aligned}
&-\frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} + p_k y_k = f_k, \qquad k = 1, \ldots, N-1;\\
&y_0 = y_N = 0;\\
&h = \frac{\pi}{N};\\
&p_k = 1 + \sin^2(\pi k h).
\end{aligned}
$$

реализуйте метод с предобуславливателем.

Решение. Реализацию см. в файле "solution.h", незамысловатая система тестирования реализована в "test3.c".

Фактически, приведенная реализация поддерживает любой предобуславливатель. В том числе, как требовалось в задаче, реализовано обращение матрицы методом Фурье.

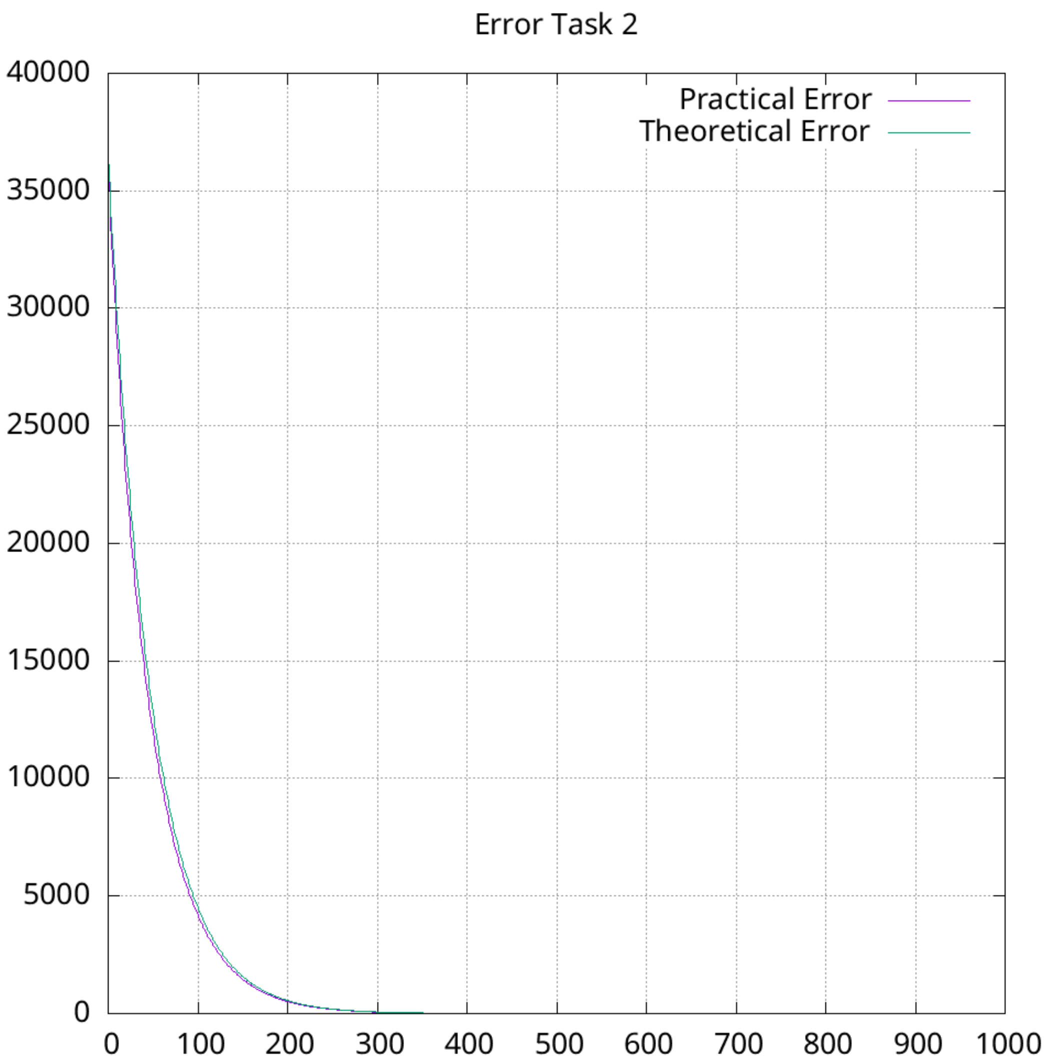

Рис. 1: Ошибка метода Ричардсона решения СЛУ, определяемого матрицей типа Фурье (задача 1).

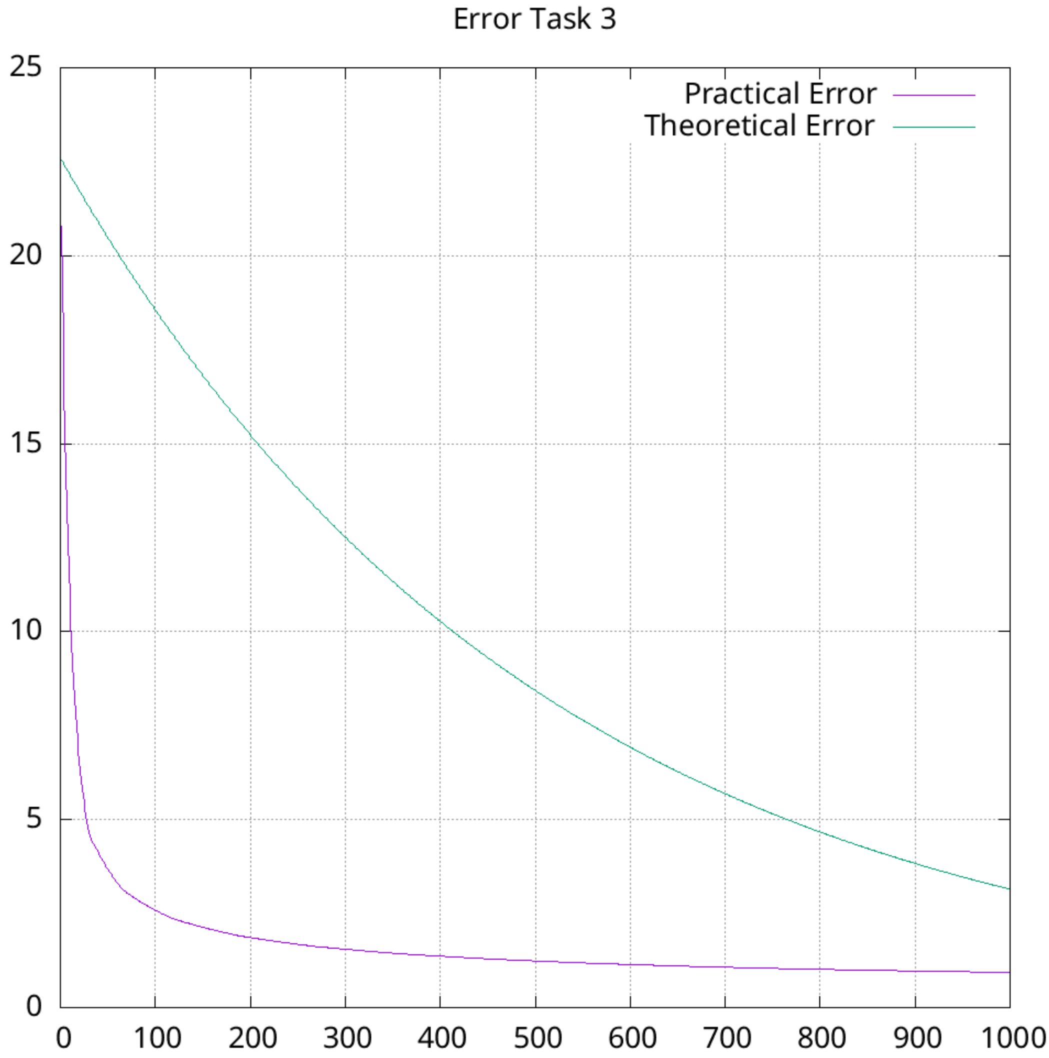

Рис. 2: Ошибка метода Ричардсона решения СЛУ, определяемого матрицей типа Фурье (задача 1).

По результатам тестирования был построен график (для вывода ./LS3 50 1000 100 out.txt, те матрица порядка 50, до 1000 итераций с $p = 100$). См. рисунок.