Практикум на ЭВМ
Токаева Александра 409
Задание 5: 1-мерный итерационный процесс для 0101

Мы хотим решить уравнение $-u'' + p(x)u = f$ с начальными условиями $u(0) = 0;\ u'(1) = 0;$
После дискретизации задачи на смещенной справа сетке с
$x_0 = 0;\ x_N = 1 + \frac{h}{2}; h = \frac{1}{N-0.5}$ получаем матрицу A $(N-1) \times (N-1)$

$$\begin{matrix} \frac{2}{h^2} + p(x_1) & -\frac{1}{h^2} & 0 \\ -\frac{1}{h^2} & \frac{2}{h^2} + p(x_k) & -\frac{1}{h^2} \\ 0 & -\frac{1}{h^2} & \boldsymbol{\frac{1}{h^2}} + p(x_{N-1}) \end{matrix}$$

Такую систему Ay=f мы методом Фурье не решим, потому что мы не знаем собственных функций и собственных значений матрицы A, потому что диагональные добавки зависят от номера строки.
Но если бы вместо $p(x_k)$ на всей диагонали стояло бы одинаковое константное число p_0 (которое мы выбираем так, чтобы оно как-то было похоже на поведение p, то такую систему Bx=g мы легко решим методом Фурье, поскольку собственные функции матрицы B — такие же, как у просто матрицы оператора Лапласа на нашей сетке, без диагональных добавок, а собственные значения — это собственные значения матрицы оператора Лапласа плюс p_0. По собственным функциям мы правую часть легко раскладываем, а именно $g = \sum_{i=1}^{N} d_i \varphi^{(i)}$, где $d_i = \frac{<g, \varphi^{(i)}>}{<\varphi^{(i)}, \varphi^{(i)}>}$
А поскольку каждая базисная функция $\varphi^{(i)}$ в процессе применения к ней матрицы B умножается $\lambda_i + p_0$, то решением системы Bx=f будет вектор $x = \sum_{i=1}^{N} \alpha_i \varphi^{(i)}$, *где* $\alpha_i = \frac{d_i}{\lambda_i + p_0}$

Внимание! Поскольку дальше мы из вектора x будем делать вектор y^k, который именно вектор значений, а не вектор коэффициентов разложения по базису из синусов, то нужно не забыть из вектора коэффициентов α сделать именно вектор значений, применив к α функцию trig_mnog (которая имеет смысл c2f);

Напомним, какие в нашей задаче собственные функции и собственные значения (вывод функций см. В отчете по 1-мерному Фурье):

$$\varphi_k^{(n)} = \sin(\pi \cdot (n - 0.5) \cdot x_k); n = 1 \dots N-1$$

n-номер функции, k-координата внутри вектора

Найдем λ_n из условия $-\frac{y_{k+1}-2y_k+y_{k-1}}{h^2} = \lambda_n \cdot y_k$: $\quad y_{k+1} - 2y_k + y_{k-1} = \sin(\pi \cdot (n-0.5) \cdot (k+1)h) - 2\sin(\pi \cdot (n-0.5) \cdot kh) + \sin(\pi \cdot (n-0.5) \cdot (k-1)h) = 2\sin(\pi \cdot (n-0.5) \cdot kh)\cos(\pi \cdot (n-0.5) \cdot h) - 2\sin(\pi \cdot (n-0.5) \cdot kh) = -2\sin(\pi \cdot (n-0.5) \cdot kh)\,(1-\cos(\pi \cdot (n-0.5) \cdot h)) = -4\sin(\pi \cdot (n-0.5) \cdot kh)\left(\sin\left(\frac{\pi \cdot (n-0.5) \cdot h}{2}\right)\right)^2 = -\lambda_n \cdot h^2 \cdot \sin\,(\pi \cdot (n-0.5) \cdot kh)$;

Отсюда $\lambda_n = \frac{4\left(\sin\left(\frac{\pi \cdot (n-0.5) \cdot h}{2}\right)\right)^2}{h^2}; n = 1 \dots N-1$

И теперь, умея решать матрицу Bx=g методом Фурье, решение исходной задачи мы будем искать как то, к чему сходится итерационный процесс

$$B\left(\frac{y^{k+1}-y^k}{\tau}\right) - Ay^k = f$$

Теория обещает, что при правильном выборе τ сходиться он будет с любого начального приближения, поэтому положим $y^0 = (0 \dots .. 0)$;
Тогда для нахождения y^{k+1} по y^k достаточно перенести Ay^k вправо, решить методом Фурье систему BZ=$f + Ay^k$, и потом выразить $y^{k+1} - y^k$ из условия

$$\frac{y^{k+1}-y^k}{\tau} = x, \text{ то есть } y^{k+1} = y^k + \tau x$$

Разберемся, какое выбрать τ.
Мы на семинаре по численным методам выяснили, что оптимальным параметром будет $\tau = \frac{2}{m+M}$, где m и M — какие-то константы, ограничивающие отношение $m \le \frac{(Ax,x)}{(Bx,x)} \le M$
На семинаре по ЭВМ мы выяснили, что

$$\frac{(Ax,x)}{(Bx,x)} = \frac{(Ax,x)}{(B^{0.5}x, B^{0.5}x)} = \frac{(B^{-0.5}AB^{-0.5}B^{0.5}x, B^{0.5}x)}{(B^{0.5}x, B^{0.5}x)} = \frac{(B^{-0.5}AB^{-0.5}y, y)}{(y,y)}$$

А чтобы ограничить такую дробь, достаточно найти минимальное и максимальное собственные значения матрицы $B^{-0.5}AB^{-0.5}$
А собственные значения такой матрицы равны $\frac{\lambda_n+p}{\lambda_n+p_0}$; минимум почти единица при n=N-1, максимум при n=1.

Гоняем итерационный процесс, пока
$\|z^k\| = \|b - Ay^k\|$ не станет $< EPS = 10^{-10}$.

Проверяем нашу программу:
1) Если $p_0 = p = const$, то просто матрицы A и B совпадают, и процесс должен сойтись за 1 шаг. В программе надо поставить p=10, p_0=p и tau=get_tau() (а не get_tau(N, mas_x), которое для случая когда p существенно зависит от x)

```
(base) MacBook-Pro-Aleksandra:task5_iter_process aleksandra$ ./a.out 0
1 10 ravnom myfunc2
Hello!
 a=0.000000 b=1.000000 N=10 is_ravnom=0 myfunc2
ravnom uzl
m=1.000000 M=1.000000 tau=1.000000 tau_new=0.666667
k=1 err=5.402451e-06
k=2 err=4.303521e-11
END: k=2
Goodbuy!
```

Ну за 2 шага сошлось, причем на 1 шаге err уже в принципе и так нулем было, так что все хорошо.

2) Меняем на $p_0 = 0.5p$. Тогда $\tau = \frac{2}{3}, q = \frac{1}{3}$, $10^{10} \sim 3^{20.959}$, поэтому должно сойтись примерно за 21 шаг.

```
(base) MacBook-Pro-Aleksandra:task5_iter_process aleksandra$ ./a.out 0
1 1000 ravnom myfunc2
Hello!
 a=0.000000 b=1.000000 N=1000 is_ravnom=0 myfunc2
ravnom uzl
m=1.000001 M=1.669577 tau=0.666667 tau_new=0.545022
k=1 err=2.078845e-01
k=2 err=6.627017e-02
k=3 err=2.258121e-02
k=4 err=7.975068e-03
k=5 err=2.848546e-03
k=6 err=1.015849e-03
k=7 err=3.598631e-04
k=8 err=1.264669e-04
k=9 err=4.410228e-05
k=10 err=1.527372e-05
k=11 err=5.258208e-06
k=12 err=1.801065e-06
k=13 err=6.142781e-07
k=14 err=2.087570e-07
k=15 err=7.073073e-08
k=16 err=2.390419e-08
k=17 err=8.061651e-09
k=18 err=2.713936e-09
k=19 err=9.120997e-10
k=20 err=3.061868e-10
k=21 err=1.026898e-10
k=22 err=3.450666e-11
END: k=22
```

```
Goodbuy!
```

3) Меняем на p(x) на $1 + x^2$, $p_0 = 0.5(pmin + pmax)$, get_tau() на get_tau(N,mas_x), получаем:

```
(base) MacBook-Pro-Aleksandra:task5_iter_process aleksandra$ ./a.out 0
1 1000 ravnom myfunc2
Hello!
 a=0.000000 b=1.000000 N=1000 is_ravnom=0 myfunc2
ravnom uzl
m=1.000000 M=1.168498 tau=0.856980 tau_new=0.631214
k=1 err=1.203557e-01
k=2 err=2.430637e-02
k=3 err=4.620355e-03
k=4 err=8.225524e-04
k=5 err=1.396809e-04
k=6 err=2.294513e-05
k=7 err=3.678897e-06
k=8 err=5.792526e-07
k=9 err=8.992859e-08
k=10 err=1.380707e-08
k=11 err=2.100690e-09
k=12 err=3.174124e-10
k=13 err=4.766504e-11
END: k=13
Goodbuy!
```