Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК4 «Программное обеспечение ЭВМ, информационные технологии»

Лабораторная работа №4

«Применение базовых методов решения ДУЧП2 параболического типа»

ДИСЦИПЛИНА: «Моделирование»

Выполнил: студент гр. ИУК4-62Б _____ (Калашников А.С.)
(подпись) (Ф.И.О.)

Проверил: _____ (Никитенко У.В.)
(подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:
- Оценка:

Калуга, 2023

Цель работы: сформировать практические навыки анализа возможностей построения и выделения наиболее важных свойств объектов моделей для моделирования и использования специализированных программных пакетов и библиотек для стандартных вычислений и визуализации результатов численного или приближенно-аналитического решения ДУЧП2 параболического типа на основе сравнения результатов.

Задачи: решить уравнение, указанное в варианте методом аппроксимации дифференциального оператора. Выбрать среду для проведения расчетов и вычислительного эксперимента. Написать программу, реализующую решение задачи. Оценить результаты расчетов. Визуализировать результаты.

Вариант №8

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left((x+1)\frac{\partial u}{\partial x}\right) - u + f(x,t),$$

$$u(x,0) = \varphi(x), \quad 0 \leqslant x \leqslant 1,$$

$$u(0,t) = \alpha(t), \quad u(1,t) + \left.\frac{\partial u}{\partial x}\right|_{x=1} = \beta(t), \ 0 \leqslant t \leqslant 0.1.$$

Используя различные разностные схемы

- явную схему порядка $O(h^2 + \tau)$ с аппроксимацией производных в граничных условиях с порядком $O(h^2)$;
- схему с весами при σ = 0, σ = 1, σ = 1/2 с аппроксимацией производных в граничных условиях с порядком $O(h^2)$.

По решению задачи должен быть представлен отчет, содержащий

1) Алгоритм решения задачи.
2) Тестирование алгоритма на решениях, для которых разностная схема точно аппроксимирует дифференциальную задачу.
3) Тестирование алгоритма, например, на решениях $u(x,t) = x^3 + t^3, sin(2t+1) * cos(2x)$, на которых разностная схема неточно аппроксимирует дифференциальную задачу.
4) Таблицы решения на «крупной» сетке независимо от шагов по t и x, с которыми строится решение, следующего вида ($N = 5, 10, 20$).
5) Таблицы, характеризующие точность решения и внутреннюю сходимость.

Ход решения с явной разностной схемой $O(h^2+\tau)$

$$\frac{du}{dt} = Lu + f_i^k$$

Алгоритм решения

Найдем Lu:

$$Lu = \frac{d}{dx}\left(p(x)\frac{du}{dx}\right) + b(x,t)\frac{du}{dx} + c(x,t)u = \frac{d}{dx}\left((x+1)\frac{du}{dx}\right) - u$$

Найдем $p(x), b(x,t), c(x,t)$:

$$p(x) = x + 1$$

$$b(x,t) = 0$$

$$c(x,t) = -1$$

Найдем значение $L_h u_i^k$:

$$L_h u_i^k u = p_{i+\frac{1}{2}}\frac{u_{i+1}^k - u_i^k}{h^2} - p_{i-\frac{1}{2}}\frac{u_i^k - u_{i-1}^k}{h^2} - u_i^k$$

Находим u_i^0

$$u_i^0 = u(x_i, 0) = \varphi(x_i)$$

Вычислим граничные условия:

$$u_0^k = \alpha(t_k)$$

$$u_N^k + \frac{3u_N^k - 4u_{N-1}^k + u_{N-2}^k}{2h} = \beta(t_k)$$

$$2hu_N^k + 3u_N^k - 4u_{N-1}^k + u_{N-2}^k = 2h\beta(t_k)$$

$$(2h+3)u_N^k = 2h\beta(t_k) + 4u_{N-1}^k - u_{N-2}^k$$

$$u_N^k = \frac{2h\beta(t_k) + 4u_{N-1}^k - u_{N-2}^k}{2h+3}$$

Запишем исходное выражение с использованием узловых значений:

$$L_h u_i^k = p_{i+\frac{1}{2}}\frac{u_{i+1}^k - u_i^k}{h^2} - p_{i-\frac{1}{2}}\frac{u_i^k - u_{i-1}^k}{h^2} - u_i^k$$

$$\frac{u_i^k - u_i^{k-1}}{\tau} = L_h u_i^{k-1} + f(x_i, t_{k-1})$$

Выразим текущее значение из предыдущих:

$$u_i^k = u_i^{k-1} + \tau\left(L_h u_i^{k-1} + f(x_i, t_{k-1})\right)$$

Для проверки точного решения используем функцию $u = x^2 + t$, тогда

$$1 = \frac{\partial}{\partial x}\big((x+1)2x\big) - x^2 - t + f(x,t)$$

$$1 = 4x + 2 - x^2 - t + f(x,t)$$

$$f(x,t) = x^2 - 4x - 1 + t$$

$$\phi(x) = x^2$$

$$\alpha(t) = t$$

$$\beta(t) = t + 3$$

При подстановке функций в программу получим 0 погрешность

h	tau	||U_{exact}-U_{h}||	||U_{2h}-U_{h}||
0.25	0.001	0.0	0.0
0.125	0.001	0.0	0.0
0.0625	0.001	0.0	0.0

Рисунок 1 Результат работы явной схемы для точной аппроксимации

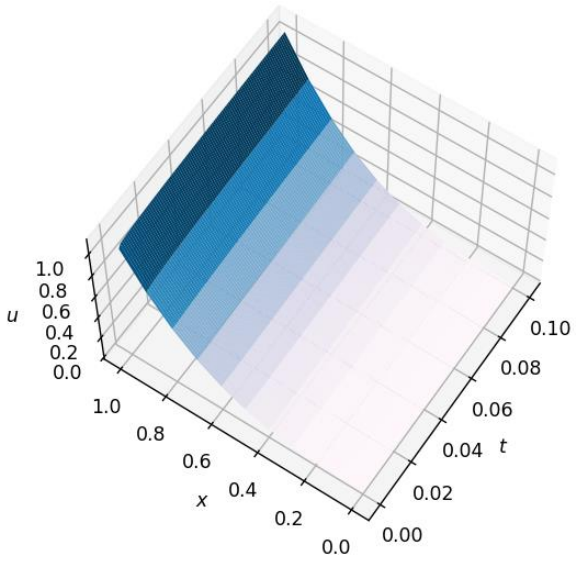


Рисунок 2 График аппроксимированной функции

Протестируем алгоритм разностной схемы на функции $u = x^3 + t^3$, тогда

$$3t^2 = \frac{\partial}{\partial x}\big((x+1)(3x^2)\big) - x^3 - t^3 + f(x,t)$$

$$f(x,t) = x^3 - 9x^2 - 6x + 3t^2 + t^3$$

$$\phi(x) = x^3$$

$$\alpha(t) = t^3$$

$$\beta(t) = 1 + t^3 + 3$$

x \ t	0.0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.001	0.001	0.001
0.25	0.016	0.016	0.017	0.017	0.018	0.019	0.02	0.021	0.023	0.024	0.025
0.5	0.125	0.126	0.126	0.128	0.131	0.133	0.135	0.138	0.14	0.142	0.144
0.75	0.422	0.422	0.428	0.433	0.437	0.441	0.444	0.447	0.45	0.452	0.454
1.0	1.0	1.018	1.025	1.03	1.034	1.037	1.04	1.043	1.045	1.048	1.05

Рисунок 3 Результат аппроксимации при $h = 0.25$

x \ t	0.0	0.004	0.008	0.012	0.016	0.02	0.023	0.027	0.031	0.035	0.039	0.043	0.047	0.051	0.055	0.058	0.062	0.066	0.07	0.074	0.078	0.082	0.086	0.09	0.094	0.097	0.101
0.5	0.125	0.125	0.125	0.125	0.125	0.125	0.126	0.126	0.126	0.126	0.127	0.127	0.127	0.127	0.128	0.128	0.128	0.128	0.129	0.129	0.129	0.129	0.13	0.13	0.13	0.131	0.131
0.625	0.244	0.244	0.244	0.245	0.245	0.245	0.245	0.246	0.246	0.246	0.247	0.247	0.247	0.248	0.248	0.248	0.249	0.249	0.249	0.249	0.25	0.25	0.25	0.251	0.251	0.251	0.252
0.75	0.422	0.422	0.423	0.423	0.423	0.424	0.424	0.425	0.425	0.426	0.426	0.426	0.427	0.427	0.427	0.428	0.428	0.428	0.429	0.429	0.429	0.43	0.43	0.43	0.43	0.431	0.431
0.875	0.67	0.671	0.672	0.673	0.673	0.674	0.674	0.675	0.675	0.676	0.676	0.676	0.677	0.677	0.677	0.678	0.678	0.678	0.679	0.679	0.679	0.68	0.68	0.68	0.681	0.681	0.681
1.0	1.002	1.004	1.005	1.005	1.006	1.007	1.007	1.008	1.008	1.008	1.009	1.009	1.009	1.01	1.01	1.01	1.011	1.011	1.011	1.011	1.012	1.012	1.012	1.013	1.013	1.013	1.013

Рисунок 4 Результат аппроксимации при $h = 0.125$

h	tau	||U_{exact}-U_{h}||	||U_{2h}-U_{h}||
0.25	0.0005	0.04962	0.0
0.125	0.0005	0.01233	0.03729
0.0625	0.0005	0.00308	0.00926

Рисунок 5 Таблица, характеризующая точность полученного решения

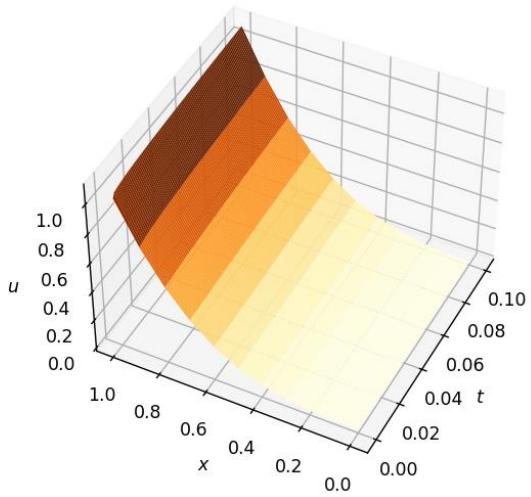


Рисунок 6 График аппроксимированной функции

Таким образом, с уменьшением шага по x при фиксированном шаге по t растёт точность аппроксимации.

Ход решения с использованием схемы с весами

$$\frac{du}{dt} = Lu + f_i^k$$

Алгоритм решения

Найдем Lu:

$$Lu = \frac{d}{dx}\left(p(x)\frac{du}{dx}\right) + b(x,t)\frac{du}{dx} + c(x,t)u = \frac{d}{dx}\left((x+1)\frac{du}{dx}\right) - u$$

Найдем $p(x), b(x,t), c(x,t)$:

$$p(x) = x + 1$$

$$b(x,t) = 0$$

$$c(x,t) = -1$$

Найдем значение $L_h u_i^k$:

$$L_h u_i^k u = p_{i+\frac{1}{2}}\frac{u_{i+1}^k - u_i^k}{h^2} - p_{i-\frac{1}{2}}\frac{u_i^k - u_{i-1}^k}{h^2} - u_i^k$$

Найдём начальные условия:

$$u_i^0 = \phi(x)$$

Вычислим граничные условия:

$$\alpha(t_k) = u_0^k$$

$$\beta(t_k) = u_N^k + \frac{u_N^k - u_{N-1}^k}{h}$$

$$h\beta(t_k) = hu_N^k + u_N^k - u_{N-1}^k$$

$$u_N^k(h+1) - u_{N-1}^k = h\beta(t_k)$$

$$u_N^k\left(\frac{1}{h}+1\right) - \frac{1}{h}u_{N-1}^k = \beta(t_k)$$

Для решения на каждом последующем слое необходимо решить систему уравнений. Составим коэффициенты для этой системы.

$$\sigma L_h u_i^k - \frac{1}{\tau}u_i^k = G_i^k, \text{где}$$

$$G_i^k = -\frac{1}{\tau}u_i^{k-1} - (1-\sigma)L_h u_i^{k-1} - f(x_i, t_k)$$

Подставим значение $L_h u_i^k$ из прошлого случая:

$$\sigma\left(p_{i+\frac{1}{2}}\frac{u_{i+1}^k - u_i^k}{h^2} - p_{i-\frac{1}{2}}\frac{u_i^k - u_{i-1}^k}{h^2} - u_i^k\right) - \frac{1}{\tau}u_i^k = G_i^k$$

$$\frac{\sigma p_{i+0.5}}{h^2}u_{i+1}^k - \left(\frac{\sigma p_{i+0.5}}{h^2} + \frac{\sigma p_{i-0.5}}{h^2} + \sigma + \frac{1}{\tau}\right)u_i^k + \frac{\sigma p_{i-0.5}}{h^2}u_{i-1}^k = G_i^k$$

Исходя из полученного уравнения и граничных условий, можно составить следующую систему:

$$\begin{matrix} & 0 & u_0^k & 0 & \\ & & \cdots & & \\ \left(\frac{\sigma p_{i+0.5}}{h^2}\right)\mathrm{u}_{i+1}^{k+1} & -\left(\frac{\sigma p_{i+0.5}}{h^2} + \frac{\sigma p_{i-0.5}}{h^2} + \sigma + \frac{1}{\tau}\right)\mathrm{u}_i^{k+1} & \left(\frac{\sigma p_{i-0.5}}{h^2}\right)\mathrm{u}_{i-1}^k = & \begin{matrix}\alpha(t_k)\\ \cdots \\ G_i^k \\ \cdots \\ \beta(t_k)\end{matrix} \\ & & \cdots & & \\ & u_{N-1}^k & (h+1)u_N^{k+1} & 0 & \end{matrix}$$

Для проверки точного решения возьмём функцию $u = x + t$, тогда

$$1 = \frac{\partial}{\partial x}\big((x+1)\big) - x - t + f(x,t)$$

$$1 = 1 - x - t + f(x,t)$$

$$f(x,t) = x + t$$

$$\phi(x) = x$$

$$\alpha(t) = t$$

$$\beta(t) = 1 + t + 1 = t + 2$$

Проверим правильность для точной аппроксимации $u = x + t$ при $\sigma = 0$.

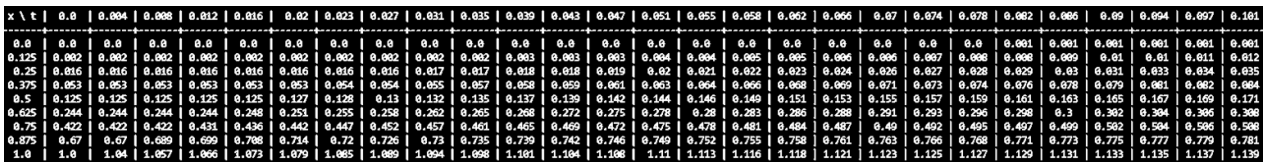
h	tau	||U_{exact}-U_{h}||	||U_{2h}-U_{h}||
0.2	0.0001	0.0	0.0
0.1	0.0001	0.0	0.0
0.05	0.0001	0.0	0.0
0.025	0.0001	0.0	0.0

Рисунок 7 Таблица, характеризующая точность решения для точной функции

Аппроксимируем функцию $u = x^3 + t^3$:
Вычислим значения u для функции $u = x^3 + t^3$при $\sigma = 0$.

x \ t	0.0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.001	0.001	0.001
0.25	0.016	0.016	0.017	0.017	0.02	0.025	0.03	0.036	0.043	0.049	0.055
0.5	0.125	0.126	0.126	0.138	0.151	0.164	0.176	0.188	0.199	0.209	0.219
0.75	0.422	0.423	0.464	0.493	0.514	0.533	0.548	0.562	0.574	0.585	0.595
1.0	1.0	1.138	1.171	1.194	1.212	1.226	1.239	1.25	1.259	1.268	1.277

Рисунок 8 Результат аппроксимации при $h = 0.25$



x \ t	0.0	0.004	0.008	0.012	0.016	0.02	0.023	0.027	0.031	0.035	0.039	0.043	0.047	0.051	0.055	0.058	0.062	0.066	0.07	0.074	0.078	0.082	0.086	0.09	0.094	0.097	0.101
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.001	0.001	0.001	0.001	0.001	0.001
0.125	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.003	0.003	0.003	0.004	0.004	0.005	0.005	0.006	0.007	0.008	0.008	0.009	0.01	0.01	0.011	0.011	0.012
0.25	0.016	0.016	0.016	0.016	0.016	0.016	0.016	0.016	0.017	0.017	0.018	0.018	0.019	0.02	0.021	0.022	0.023	0.024	0.026	0.027	0.028	0.029	0.03	0.031	0.033	0.034	0.035
0.375	0.053	0.053	0.053	0.053	0.053	0.053	0.054	0.054	0.055	0.057	0.058	0.059	0.061	0.063	0.064	0.066	0.068	0.069	0.071	0.073	0.074	0.076	0.078	0.079	0.081	0.082	0.084
0.5	0.125	0.125	0.125	0.125	0.125	0.127	0.128	0.13	0.132	0.135	0.137	0.139	0.142	0.144	0.146	0.149	0.151	0.153	0.155	0.157	0.159	0.161	0.163	0.165	0.167	0.169	0.171
0.625	0.244	0.244	0.244	0.244	0.248	0.251	0.255	0.258	0.262	0.265	0.268	0.272	0.275	0.278	0.28	0.283	0.286	0.288	0.291	0.293	0.296	0.298	0.3	0.302	0.304	0.306	0.308
0.75	0.422	0.422	0.422	0.431	0.436	0.442	0.447	0.452	0.457	0.461	0.465	0.469	0.472	0.475	0.478	0.481	0.484	0.487	0.49	0.492	0.495	0.497	0.499	0.502	0.504	0.506	0.508
0.875	0.67	0.67	0.689	0.699	0.708	0.714	0.72	0.726	0.73	0.735	0.739	0.742	0.746	0.749	0.752	0.755	0.758	0.761	0.763	0.766	0.768	0.771	0.773	0.775	0.777	0.779	0.781
1.0	1.0	1.04	1.057	1.066	1.073	1.079	1.085	1.089	1.094	1.098	1.101	1.104	1.108	1.11	1.113	1.116	1.118	1.121	1.123	1.125	1.127	1.129	1.131	1.133	1.135	1.137	1.139

Рисунок 9 Результат аппроксимации при $h = 0.1$

h	tau	||U_{exact}-U_{h}||	||U_{2h}-U_{h}||
0.25	0.0005	0.2791	0.0
0.125	0.0005	0.1378	0.1413
0.0625	0.0005	0.0683	0.0695

Рисунок 11 Таблица, характеризующая точность полученного решения

Проверим правильность для точной аппроксимации $u = x + t$ при $\sigma = 1$.

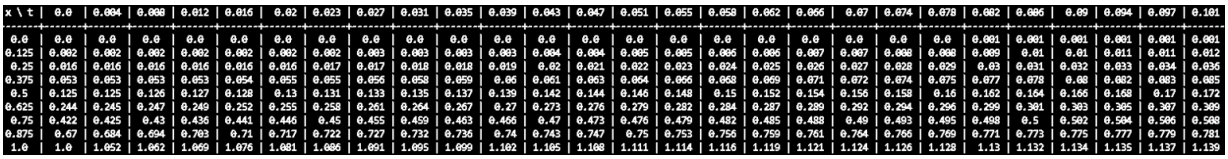
h	tau	||U_{exact}-U_{h}||	||U_{2h}-U_{h}||
0.25	0.0005	0.0	0.0
0.125	0.0005	0.0	0.0
0.0625	0.0005	0.0	0.0

Рисунок 12 Таблица, характеризующая точность решения для точной функции

Аппроксимируем функцию $u = x^3 + t^3$:
Вычислим значения u для функции $u = x^3 + t^3$при $\sigma = 1$.

x \ t	0.0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.001	0.001	0.001
0.25	0.016	0.017	0.02	0.023	0.027	0.032	0.037	0.042	0.047	0.052	0.057
0.5	0.125	0.131	0.141	0.151	0.162	0.173	0.184	0.194	0.203	0.212	0.221
0.75	0.422	0.455	0.481	0.503	0.522	0.538	0.552	0.565	0.576	0.587	0.596
1.0	1.0	1.164	1.185	1.202	1.217	1.23	1.242	1.252	1.261	1.27	1.277

Рисунок 13 Результат аппроксимации при $h = 0.25$



x \ t	0.0	0.004	0.008	0.012	0.016	0.02	0.023	0.027	0.031	0.035	0.039	0.043	0.047	0.051	0.055	0.058	0.062	0.066	0.07	0.074	0.078	0.082	0.086	0.09	0.094	0.097	0.101
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.001	0.001	0.001	0.001	0.001	0.001
0.125	0.002	0.002	0.002	0.002	0.002	0.002	0.003	0.003	0.003	0.003	0.003	0.004	0.004	0.005	0.005	0.006	0.006	0.007	0.007	0.008	0.008	0.009	0.01	0.01	0.011	0.011	0.012
0.25	0.016	0.016	0.016	0.016	0.016	0.016	0.017	0.017	0.018	0.018	0.019	0.02	0.021	0.022	0.023	0.024	0.025	0.026	0.027	0.028	0.029	0.03	0.031	0.032	0.033	0.034	0.036
0.375	0.053	0.053	0.053	0.053	0.054	0.055	0.055	0.056	0.058	0.059	0.06	0.061	0.063	0.064	0.066	0.068	0.069	0.071	0.072	0.074	0.075	0.077	0.078	0.08	0.082	0.083	0.085
0.5	0.125	0.125	0.126	0.127	0.128	0.13	0.131	0.133	0.135	0.137	0.139	0.142	0.144	0.146	0.148	0.15	0.152	0.154	0.156	0.158	0.16	0.162	0.164	0.166	0.168	0.17	0.172
0.625	0.244	0.245	0.247	0.249	0.252	0.255	0.258	0.261	0.264	0.267	0.27	0.273	0.276	0.279	0.282	0.284	0.287	0.289	0.292	0.294	0.296	0.299	0.301	0.303	0.305	0.307	0.309
0.75	0.422	0.425	0.43	0.436	0.441	0.446	0.45	0.455	0.459	0.463	0.466	0.47	0.473	0.476	0.479	0.482	0.485	0.488	0.49	0.493	0.495	0.498	0.5	0.502	0.504	0.506	0.508
0.875	0.67	0.684	0.694	0.703	0.71	0.717	0.722	0.727	0.732	0.736	0.74	0.743	0.747	0.75	0.753	0.756	0.759	0.761	0.764	0.766	0.769	0.771	0.773	0.775	0.777	0.779	0.781
1.0	1.0	1.052	1.062	1.069	1.076	1.081	1.086	1.091	1.095	1.099	1.102	1.105	1.108	1.111	1.114	1.116	1.119	1.121	1.124	1.126	1.128	1.13	1.132	1.134	1.135	1.137	1.139

Рисунок 14 Результат аппроксимации при $h = 0.125$

h	tau	||U_{exact}-U_{h}||	||U_{2h}-U_{h}||
0.25	0.0005	0.2791	0.0
0.125	0.0005	0.1378	0.1413
0.0625	0.0005	0.0683	0.0695

Рисунок 15 Таблица, характеризующая точность решения для аппроксимируемой функции

Проверим правильность для точной аппроксимации $u = x + t$ при $\sigma = 0.5$.

h	tau	||U_{exact}-U_{h}||	||U_{2h}-U_{h}||
0.25	0.0005	0.0	0.0
0.125	0.0005	0.0	0.0
0.0625	0.0005	0.0	0.0

Рисунок 16 Результат аппроксимации при $h = 0.25$

Аппроксимируем функцию $u = x^3 + t^3$:
Вычислим значения u для функции $u = x^3 + t^3$при $\sigma = 0.5$.

x \ t	0.0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.001	0.001	0.001
0.25	0.016	0.016	0.018	0.02	0.024	0.029	0.034	0.039	0.045	0.051	0.056
0.5	0.125	0.128	0.134	0.145	0.157	0.169	0.18	0.191	0.201	0.211	0.22
0.75	0.422	0.44	0.472	0.497	0.518	0.535	0.55	0.563	0.575	0.586	0.596
1.0	1.0	1.152	1.178	1.198	1.214	1.228	1.24	1.251	1.26	1.269	1.277

Рисунок 17 Результат аппроксимации при $h = 0.25$

x \ t	0.0	0.004	0.008	0.012	0.016	0.02	0.023	0.027	0.031	0.035	0.039	0.043	0.047	0.051	0.055	0.058	0.062	0.066	0.07	0.074	0.078	0.082	0.086	0.09	0.094	0.097	0.101
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.001	0.001	0.001	0.001	0.001	0.001
0.125	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.003	0.003	0.003	0.003	0.004	0.004	0.005	0.005	0.006	0.006	0.007	0.007	0.008	0.008	0.009	0.01	0.01	0.011	0.011	0.012
0.25	0.016	0.016	0.016	0.016	0.016	0.016	0.017	0.017	0.018	0.018	0.019	0.02	0.021	0.022	0.023	0.024	0.025	0.026	0.027	0.028	0.029	0.03	0.031	0.032	0.033	0.034	0.036
0.375	0.053	0.053	0.053	0.053	0.054	0.055	0.055	0.056	0.058	0.059	0.06	0.061	0.063	0.064	0.066	0.068	0.069	0.071	0.072	0.074	0.075	0.077	0.078	0.08	0.082	0.083	0.085
0.5	0.125	0.125	0.126	0.127	0.128	0.13	0.131	0.133	0.135	0.137	0.139	0.142	0.144	0.146	0.148	0.15	0.152	0.154	0.156	0.158	0.16	0.162	0.164	0.166	0.168	0.17	0.172
0.625	0.244	0.245	0.247	0.249	0.252	0.255	0.258	0.261	0.264	0.267	0.27	0.273	0.276	0.279	0.282	0.284	0.287	0.289	0.292	0.294	0.296	0.299	0.301	0.303	0.305	0.307	0.309
0.75	0.422	0.425	0.43	0.436	0.441	0.446	0.45	0.455	0.459	0.463	0.466	0.47	0.473	0.476	0.479	0.482	0.485	0.488	0.49	0.493	0.495	0.498	0.5	0.502	0.504	0.506	0.508
0.875	0.67	0.684	0.694	0.703	0.71	0.717	0.722	0.727	0.732	0.736	0.74	0.743	0.747	0.75	0.753	0.756	0.759	0.761	0.764	0.766	0.769	0.771	0.773	0.775	0.777	0.779	0.781
1.0	1.0	1.052	1.062	1.069	1.076	1.081	1.086	1.091	1.095	1.099	1.102	1.105	1.108	1.111	1.114	1.116	1.119	1.121	1.124	1.126	1.128	1.13	1.132	1.134	1.135	1.137	1.139

Рисунок 18 Результат аппроксимации при $h = 0.125$

h	tau	||U_exact-U_h||	||U_2h-U_h||
0.25	0.0005	0.2791	0.0
0.125	0.0005	0.1378	0.1413
0.0625	0.0005	0.0683	0.0695

Рисунок 19 Таблица, характеризующая точность решения для аппроксимируемой функции

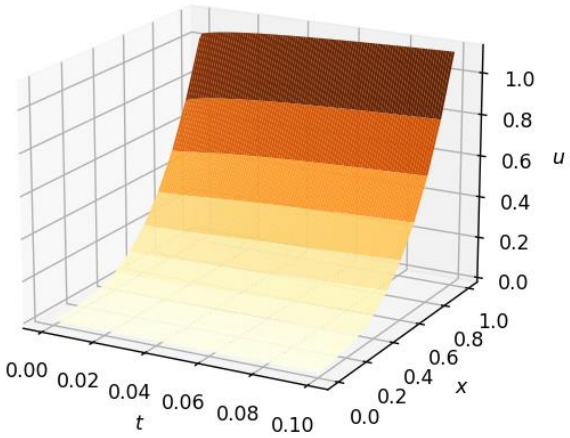


Рисунок 20 График аппроксимированной функции

Вывод: в ходе выполнения работы были сформированы практические навыки анализа возможностей построения и выделения наиболее важных свойств объектов моделей для моделирования и использования специализированных программных пакетов и библиотек для стандартных вычислений и визуализации результатов численного или приближенно-аналитического решения ДУЧП2 параболического типа на основе сравнения результатов.

ПРИЛОЖЕНИЯ

Листинг:

Задание №1

```
import numpy as np
import matplotlib.pyplot as plt
from prettytable import PrettyTable

def f(x, t):
    #return x**2 - 4*x - 1 + t
    return pow(x, 3) - 9 * pow(x, 2) - 6 * x + 3 * pow(t, 2) + pow(t, 3)

p = lambda x: x + 1

def lu(u: np.array, x: np.linspace, t: np.linspace,
       i, k, h):
    return p(x[i] + h/2) * (u[i + 1, k] - u[i, k]) / pow(h, 2) - (
        p(x[i] - h/2) * (u[i, k] - u[i - 1, k]) / pow(h, 2)) - u[i, k]


def solve(h, tau):
    x_min = 0
    x_max = 1
    xs = np.arange(x_min, x_max + h, h)
    n_x = len(xs)

    t_min = 0
    t_max = 0.1
    ts = np.arange(t_min, t_max + tau, tau)
    n_t = len(ts)

    #phi = lambda x, t: x**2 + t
    phi = lambda x, t: pow(x, 3)
    #alpha = lambda t: t
    alpha = lambda t: pow(t, 3)
    #beta = lambda t: 3 + t
    beta = lambda t: 1 + pow(t, 3) + 3

    U = np.zeros((n_x, n_t))
    U[:, 0] = [phi(x, t_min) for x in xs]

    U[0, 0] = alpha(ts[0])
    U[-1, 0] = (2*h*beta(ts[0]) + 4*U[-2, 0] - U[-3, 0])/(2*h+3)

    for k in range(0, n_t - 1):
        for i in range(1, n_x - 1):
            U[i, k + 1] = U[i, k] + tau * (lu(U, xs, ts, i, k, h) +
                                           f(xs[i], ts[k]))

        U[0, k + 1] = alpha(ts[k+1])
        U[-1, k + 1] = (2*h*beta(ts[k+1]) + 4*U[-2, k+1] - U[-3,
k+1])/(2*h+3)

    return [xs, ts, U]

def makeTableFromResult(xs, ts, U):
    table = PrettyTable()
```

```
    ts = ts.round(3)
    xs = xs.round(3)
    U = U.round(3)
    table.add_column("x \ t", xs)
    for k in range(len(ts)):
        table.add_column(f"{ts[k]}", U[:, k])
    return table
[xs, ts, U] = solve(0.0625, 0.01)

print("Результат:")
print(makeTableFromResult(xs, ts, U))

fig = plt.figure()
ax = plt.axes(projection='3d')
X, Y = np.meshgrid(ts, xs)
ax.plot_surface(X, Y, U, rstride=1, cstride=1,
    cmap='viridis', edgecolor='none')
ax.set_xlabel('$t$')
ax.set_ylabel('$x$')
ax.set_zlabel('$u$')
plt.show()

def makeTableFromStep(hs, taus, exact_diff, diff):
    table = PrettyTable()
    table.add_column("h", hs)
    table.add_column("tau", taus)
    table.add_column("||U_{exact}-U_{h}||", diff)
    table.add_column("||U_{2h}-U_{h}}||", exact_diff)
    return table

h = 0.25
tau = 0.001

hs = []
taus = []
exact_diff = []
last_diff = []

[_, _, last_u] = solve(h, tau)
last_u = last_u[0::2]

for i in range(3):
    [xs, ts, U] = solve(h, tau)

    u = lambda t, x: x**3 + t**3
    U_exact = np.array([[u(t, x) for t in ts] for x in xs])

    hs.append(h)
    taus.append(tau)
    exact_diff.append(np.amax(np.abs(U - U_exact)))
    last_diff.append(np.amax(np.abs(last_u - U[0::2])))

    h /= 2
    last_u = U

hs = np.array(hs).round(5)
taus = np.array(taus).round(5)
exact_diff = np.array(exact_diff).round(5)
last_diff = np.array(last_diff).round(5)
```

```
print(makeTableFromStep(hs, taus, last_diff, exact_diff))
```

Задание №2

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits import mplot3d
import scipy.linalg as la
from prettytable import PrettyTable
p = lambda x: x + 1
def f(x, t):
    # return x+t
    return pow(x, 3) - 9 * pow(x, 2) - 6 * x + 3 * pow(t, 2) + pow(t, 3)
def lu(u: np.array, x: np.linspace, t: np.linspace,
       i, k, h):
    return p(x[i] + h/2) * (u[i + 1, k] - u[i, k]) / pow(h, 2) - \
        p(x[i] - h/2) * (u[i, k] - u[i - 1, k]) / pow(h, 2) - u[i, k]

def solve(h, tau, sigma):
    x_min = 0
    x_max = 1
    xs = np.arange(x_min, x_max + h, h)
    n_x = len(xs)

    t_min = 0
    t_max = 0.1
    ts = np.arange(t_min, t_max + tau, tau)
    n_t = len(ts)

    # phi = lambda x: x
    # alpha = lambda t: t
    # beta = lambda t: 2 + t

    phi = lambda x: pow(x, 3)
    alpha = lambda t: pow(t, 3)
    beta = lambda t: 4 + pow(t, 3)

    U = np.zeros((n_x, n_t))
    G = np.zeros((n_x, n_t))
    U[:, 0] = [phi(x) for x in xs]
    print(U[:, 0])
    A = np.zeros((n_x - 1))
    B = np.zeros((n_x))
    C = np.zeros((n_x - 1))

    for k in range(1, n_t):
        for i in range(1, n_x - 1):
            G[i, k] = - U[i, k - 1]/tau \
                - (1-sigma) * lu(U, xs, ts, i, k - 1, h) \
                - f(xs[i], ts[k])
            A[i - 1] = sigma * p(xs[i] - h/2) / pow(h, 2)
            B[i] = - (sigma * p(xs[i] + h/2) / pow(h, 2) + \
                sigma * p(xs[i] - h/2) / pow(h, 2) + sigma + 1 / tau)
            C[i] = sigma * p(xs[i] + h/2) / pow(h, 2)

        B[0] = 1
        C[0] =  0
```

```
        A[-1] = -1
        B[-1] = h + 1

        G[0, k] = alpha(ts[k])
        G[-1, k] =  h * beta(ts[k])

        matrix = np.array([[0, *C], B, [*A, 0]])

        U[:, k] = la.solve_banded((1,1), matrix, G[:, k])

    return [xs, ts, U]

def makeTableFromResult(xs, ts, U):
    table = PrettyTable()
    ts = ts.round(4)
    xs = xs.round(4)
    U = U.round(5)
    table.add_column("x \ t", xs)
    for k in range(len(ts)):
        table.add_column(f"{ts[k]}", U[:, k])
    return table

sigma = 0.5
[xs, ts, U] = solve(0.1, 0.001, sigma)


fig = plt.figure()
ax = plt.axes(projection='3d')
X, Y = np.meshgrid(ts, xs)
ax.plot_surface(X, Y, U, rstride=1, cstride=1,
    cmap='viridis', edgecolor='none')
ax.set_xlabel('$t$')
ax.set_ylabel('$x$')
ax.set_zlabel('$u$')
plt.show()

def makeTableFromStep(hs, taus, diff, exact_diff):
    table = PrettyTable()
    table.add_column("h", hs)
    table.add_column("tau", taus)
    table.add_column("||U_{exact}-U_{h}||", diff)
    table.add_column("||U_{2h}-U_{h}}||", exact_diff)
    return table

def makeTableFromResult(xs, ts, U):
    table = PrettyTable()
    ts = ts.round(3)
    xs = xs.round(3)
    U = U.round(3)
    table.add_column("x \ t", xs)
    for k in range(len(ts)):
        table.add_column(f"{ts[k]}", U[:, k])
    return table


[xs, ts, U] = solve(0.25, 0.01, sigma)

print("Результат:")
print(makeTableFromResult(xs, ts, U))
```

```
h = 0.25
tau = 0.0005

hs = []
taus = []
exact_diff = []
last_diff = []

[_, _, last_u] = solve(h, tau, sigma)
last_u = last_u[0::2]

for i in range(3):
    [xs, ts, U] = solve(h, tau, sigma)

    u = lambda t, x: x**3 + t**3
    U_exact = np.array([[u(t, x) for t in ts] for x in xs])

    hs.append(h)
    taus.append(tau)
    exact_diff.append(np.amax(np.abs(U - U_exact)))
    last_diff.append(np.amax(np.abs(last_u - U[0::2])))

    h /= 2
    last_u = U

hs = np.array(hs).round(5)
taus = np.array(taus).round(5)
exact_diff = np.array(exact_diff).round(4)
last_diff = np.array(last_diff).round(4)

print(makeTableFromStep(hs, taus, exact_diff, last_diff))
```