

Санкт-Петербургский политехнический университет
Высшая школа прикладной математики и вычислительной физики, ФизМех

Направление подготовки
«01.03.02 Прикладная математика и информатика»

Отчет по лабораторной работе № 2
тема "Исследование разностных схем для параболических уравнений"
дисциплина "Конечно-разностные и сеточные методы"

Выполнил студент гр. 5030102/90101

М.К.Турченко

Преподаватель:

Б.С.Григорьев

Санкт-Петербург

2022

Оглавление

Постановка задачи	3
Разностные схемы	4
Явная схема	5
Неявная схема	6
Симметричная схема	7
Численная реализация	8
Результаты вычислительного эксперимента	9
Явная схема	9
Неявная схема	11
Симметричная схема	12
Сравнение эффективности схем	13
Графики точного и численных решений	14
Вывод	16

Постановка задачи

Необходимо решить параболическое уравнение:

$$\begin{aligned}\frac{\partial u}{\partial t} - (1+x) \frac{\partial^2 u}{\partial x^2} \\ = -4e^{-2t}x^2 \sin(\pi x) \\ - e^{-2t}(x+1)(2\pi(e^{2t}+4x)\cos(\pi x) \\ - (\pi^2 e^{2t}x + 2\pi^2 x^2 - 4) \sin(\pi x)), t \in [0, T], x \in (0, 1)\end{aligned}$$

со следующими граничными условиями:

$$\begin{cases} u'(0, t) = 0 \\ u(1, t) = 0 \end{cases}$$

и с начальным условием $u(x, 0) = x(2x + 1) \sin(\pi x)$;

используя три разностные схемы (явная, неявная, симметричная).

Для явной схемы найдем максимально большие шаги схемы τ_0, h_0 и проведем вычисления. Затем покажем неустойчивость решения, положив $\tau = \tau_0 + 7.75 * 10^{-5}$.

Для неявной и симметричной схем, поскольку они безусловно устойчивые, будем искать максимально большой шаг по времени τ такой, что $\varepsilon(\tau) \leq \varepsilon_0$, где ε_0 – нужная точность решения.

Также сравним эффективность трех схем по времени работы программы, необходимых для достижения заданной точности.

Точное решение: $u^*(x, t) = (2xe^{-2t} + 1)(x \sin(\pi x))$

Разностные схемы

Введем сетку на нашей области:

$$\Omega_{h\tau} = \{(x_i, t_n) | x_i = a + ih, i = 0 \dots M, h = \frac{b-a}{M}; t_n = n\tau, n = 0 \dots N, \tau = \frac{T}{N}\}$$

На этой сетке будем аппроксимировать производные согласно выбранной схеме.

Воспользовавшись начальными и граничными условиями, можем положить:

$$v_i^0 = x_i(2x_i + 1) \sin(\pi x_i), i = 0 \dots M$$

$$v_M^n = 0, n = 0 \dots N$$

Значение на левой границе v_0^n , поскольку оно содержит производную, будем находить, используя аппроксимационные схемы второго порядка.

Явная схема

$$\frac{v_i^{n+1} - v_i^n}{\tau} - \rho_i \frac{v_{i+1}^n - 2v_i^n + v_{i-1}^n}{h^2} = f_i^n, i = 1 \dots M - 1, n = 0 \dots N - 1$$

Отсюда получаем явную формулу для v_i^{n+1} :

$$v_i^{n+1} = \frac{\tau \rho_i}{h^2} (v_{i+1}^n - 2v_i^n + v_{i-1}^n) + v_i^n + \tau f_i^n, i = 1 \dots M - 1, n = 0 \dots N - 1.$$

Значение v_0^{n+1} получим, сделав левостороннюю аппроксимацию производной схемой второго порядка:

$$\frac{-3v_0^{n+1} + 4v_1^{n+1} - v_2^{n+1}}{2h} = 0 \rightarrow v_0^{n+1} = \frac{4v_1^{n+1} - v_2^{n+1}}{3}, n = 0 \dots N - 1$$

Данная схема не является абсолютно устойчивой. Условие устойчивости явной схемы: $\tau \leq \frac{h^2}{2\rho_{max}}$, где τ – величина шага по времени, h – величина шага по координате, ρ_{max} – максимальное значение числового коэффициента перед второй производной по координате искомой функции.

Найдем τ_0 – максимально возможный шаг по времени:

$$\rho = (1 + x), x \in [0, 1] \rightarrow \rho_{max} = 2$$

$$\tau \leq \frac{h^2}{2\rho_{max}} \rightarrow \tau_0 = \frac{h^2}{4}$$

Неявная схема

$$\frac{v_i^{n+1} - v_i^n}{\tau} - \rho_i \frac{v_{i+1}^{n+1} - 2v_i^{n+1} + v_{i-1}^{n+1}}{h^2} = f_i^{n+1}, i = 1 \dots M - 1, n = 0 \dots N - 1$$

$$v_{i+1}^{n+1}(-\rho_i \tau) + v_i^{n+1}(2\rho_i \tau + h^2) + v_{i-1}^{n+1}(-\rho_i \tau) = h^2(v_i^n + \tau f_i^{n+1}) \quad (1)$$

Получаем СЛАУ с почти трехдиагональной матрицей. Чтобы получить СЛАУ с трехдиагональной матрицей, подставим $i=1$ в уравнение (1) и аппроксимируем производную на левом конце отрезка схемой второго порядка. Тогда получим систему, из которой исключим v_2^{n+1} , и получившееся выражение соотношение между v_0^{n+1} и v_1^{n+1} будет недостающим уравнением СЛАУ.

$$v_2^{n+1}(-\rho_1 \tau) + v_1^{n+1}(2\rho_1 \tau + h^2) + v_0^{n+1}(-\rho_1 \tau) = h^2(v_1^n + \tau f_1^{n+1});$$

$$-3v_0^{n+1} + 4v_1^{n+1} - v_2^{n+1} = 0;$$

Тогда:

$v_0^{n+1}(2\rho_1 \tau) + v_1^{n+1}(-2\rho_1 \tau + h^2) = h^2(v_1^n + \tau f_1^{n+1})$ – недостающее уравнение для трехдиагональной СЛАУ. Эту СЛАУ легко можно решить методом прогонки.

Симметричная схема

$$\frac{v_i^{n+1} - v_i^n}{\tau} - \frac{\rho_i}{2h^2} \left((v_{i+1}^{n+1} - 2v_i^{n+1} + v_{i-1}^{n+1}) + (v_{i+1}^n - 2v_i^n + v_{i-1}^n) \right) = f_i^{n+\frac{1}{2}},$$

$$i = 1 \dots M - 1, n = 0 \dots N-1$$

$$v_{i+1}^{n+1}(-\rho_i\tau) + v_i^{n+1}(2\rho_i\tau + 2h^2) + v_{i-1}^{n+1}(-\rho_i\tau) = h^2 \left(v_i^n + \tau f_i^{n+\frac{1}{2}} \right), \quad (2)$$

$$i = 1 \dots M - 1, n = 0 \dots N-1$$

Также получаем почти трехдиагональную СЛАУ. Точно также получим недостающее уравнение для восстановления трехдиагональности матрицы системы: подставим $i=1$ в уравнение (2) и аппроксимируем производную на левом конце схемой второго порядка. Затем исключим из получившейся системы v_2^{n+1} :

$$v_2^{n+1}(-\rho_1\tau) + v_1^{n+1}(2\rho_1\tau + 2h^2) + v_0^{n+1}(-\rho_1\tau) = h^2 \left(v_1^n + \tau f_1^{n+\frac{1}{2}} \right);$$

$$-3v_0^{n+1} + 4v_1^{n+1} - v_2^{n+1} = 0;$$

Тогда:

$$v_0^{n+1}(2\rho_1\tau) + v_1^{n+1}(-2\rho_1\tau + 2h^2) = h^2 \left(v_1^n + \tau f_1^{n+\frac{1}{2}} \right) - \text{недостающее}$$

уравнение для восстановления трехдиагональности системы. Эту СЛАУ также будем решать методом прогонки.

Численная реализация

Программа была реализована на языке программирования C++ в среде разработки Microsoft Visual Studio 2019.

`class Equation` – основной класс программы. Содержит все поля и методы, необходимые для построения схемы

Сигнатура метода	Описание метода	Входные параметры	Возвращаемое значение
<code>void TridiagonalMatrixAlgorithm (double* B, double* C, double* D, double* F, int n)</code>	Решает СЛАУ с трехдиагональной матрицей	Размерность системы, массивы главной диагонали и диагоналей выше и ниже главной, массив правой части	Вектор решения СЛАУ через поле класса <code>Equation</code>
<code>Equation(int M, int N)</code>	Конструктор класса	Размер сетки	-
<code>~Equation()</code>	Деструктор класса	-	-
<code>double htNorm()</code>	Возвращает норму ошибки численного решения $\ \cdot\ _{ht}$	Численное решение через поле класса <code>Equation</code>	Норму вектора ошибки
<code>void ForwardMethod()</code>	Решает параболическое уравнение с помощью явной схемы	Сетка на области, НУ, ГУ через поле класса <code>Equation</code>	Матрицу решения параболического уравнения
<code>void BackwardMethod()</code>	Решает параболическое уравнение с помощью неявной схемы	Сетка на области, НУ, ГУ через поле класса <code>Equation</code>	Матрицу решения параболического уравнения
<code>void SymmetricMethod()</code>	Решает параболическое уравнение с помощью симметричной схемы	Сетка на области, НУ, ГУ через поле класса <code>Equation</code>	Матрицу решения параболического уравнения
<code>int main()</code>	Основной метод программы, выполняющий численный расчет	-	0, если программа завершена успешно. Иначе код ошибки

Результаты вычислительного эксперимента

Явная схема

Построим график зависимости нормы вектора ошибки $\|\cdot\|_h$ от времени t при разных шагах по времени. Сначала положим $\tau_0 = \frac{h^2}{4}$ – максимально возможный шаг для устойчивости вычислительной схемы, затем возьмем $\tau = \tau_0 + 7.75 \cdot 10^{-5}$, чтобы продемонстрировать неустойчивость явной схемы.

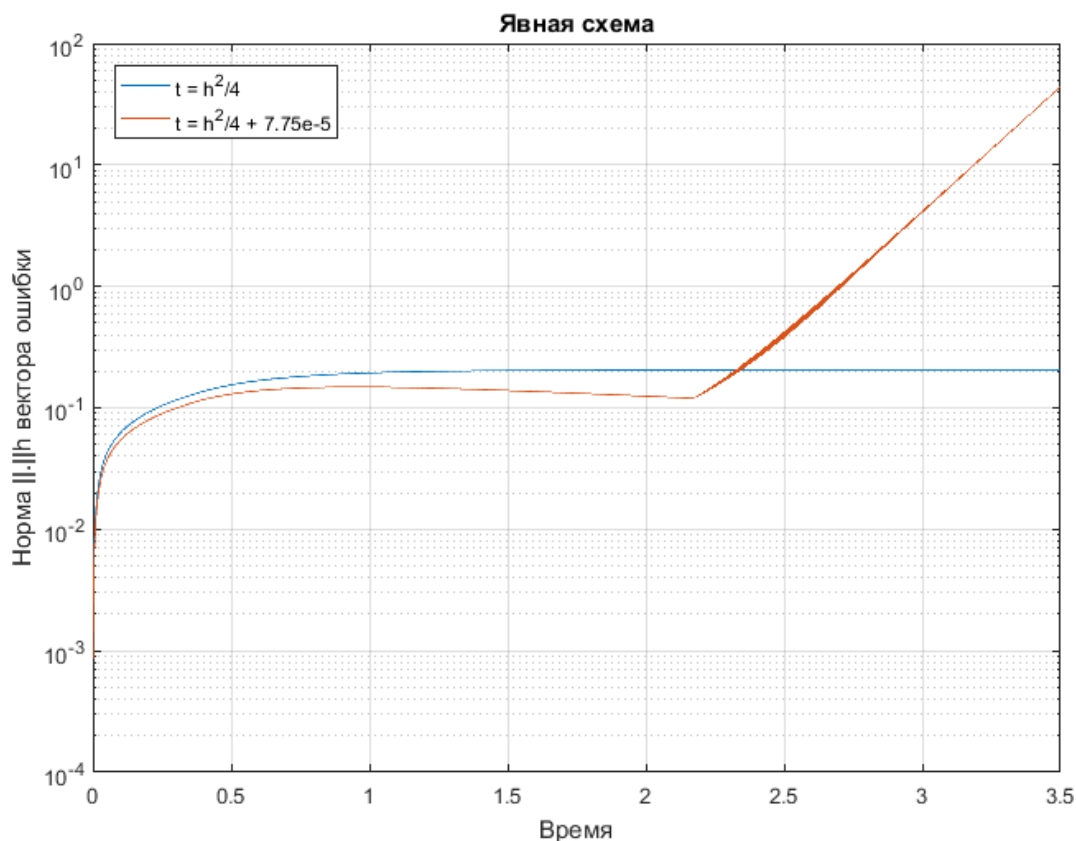


Рис. 1 Погрешность численного решения явной схемы

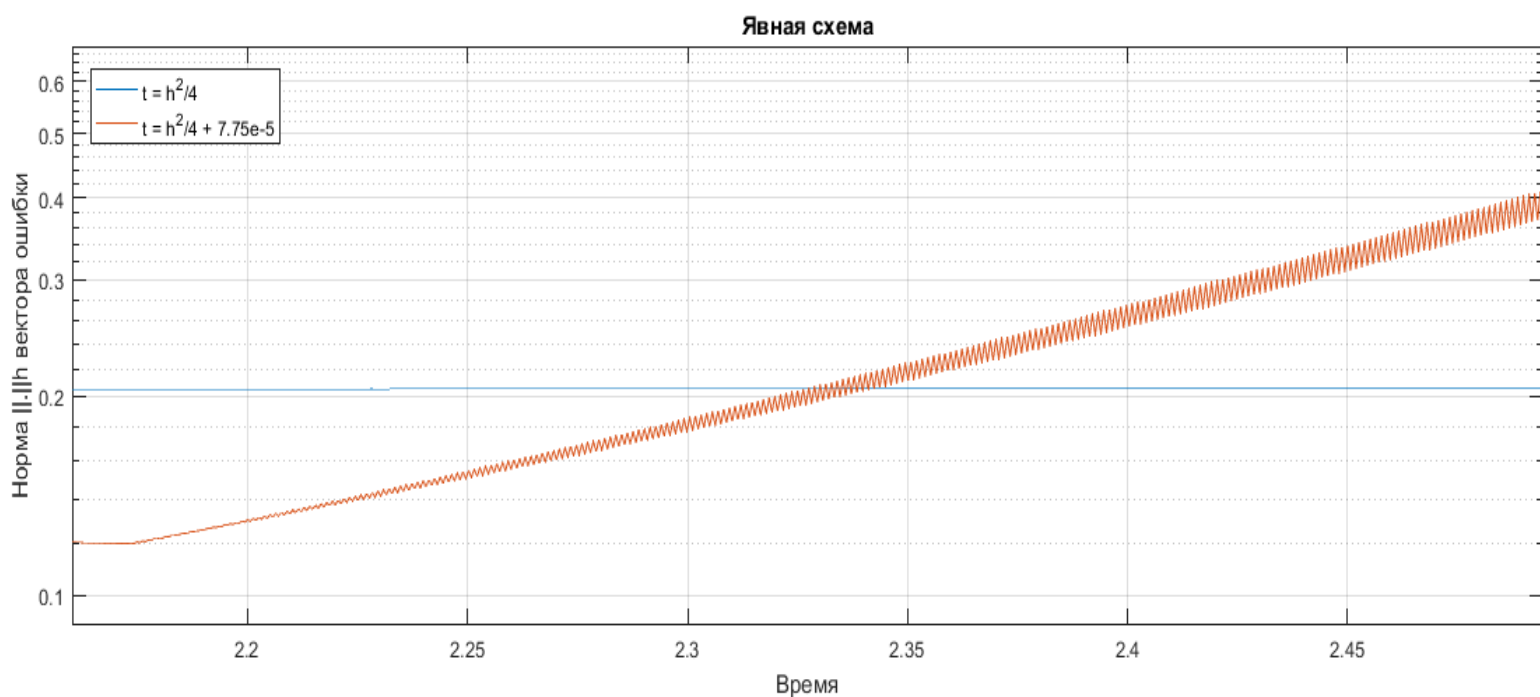


Рис. 2 Развитие неустойчивости численного решения с помощью явной схемы

На графиках видно, что даже такое малое превышение шага по времени, как $7.75 \cdot 10^{-5}$, заставляет явную схему расходиться. Неустойчивость явной схемы демонстрируется колебаниями графика нормы вектора ошибки в случае превышения допустимого шага по времени.

Неявная схема

Поскольку неявная и симметричная схемы абсолютно устойчивые, то исследуем можно ли увеличить шаг по времени τ так, чтобы ошибка решения не была больше допустимой ($\varepsilon(\tau) \leq \varepsilon_0 = 0.16$).

Построим график зависимости $\|\cdot\|_{h\tau}$ нормы вектора ошибки от величины шага τ . Будем начинать с τ_0 , пока ошибка не достигнет ε_0 .

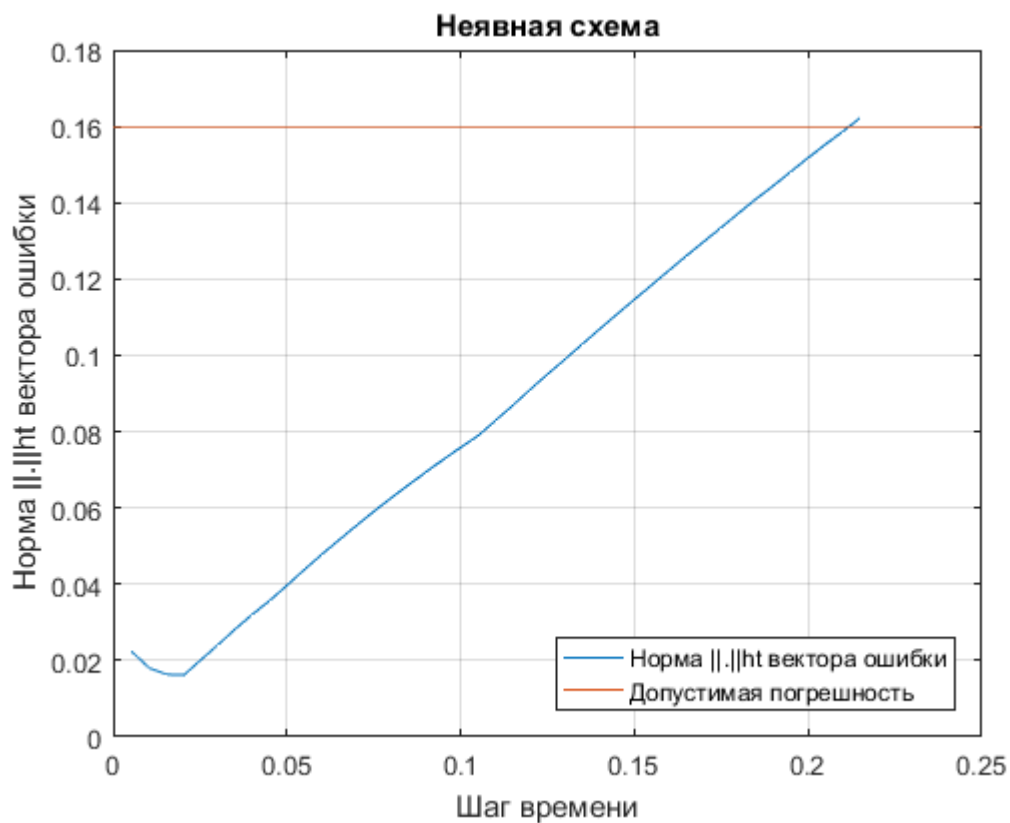


Рис. 3 Нахождение максимального шага по времени для неявной схемы

Из графика видно, что мы смогли увеличить шаг по времени с $\tau_0 = 5 * 10^{-4}$ до $\tau = 0.2$, при этом точность решения осталась нужной.

Симметричная схема

Проведем для симметричной схемы то же исследование, что и для неявной:

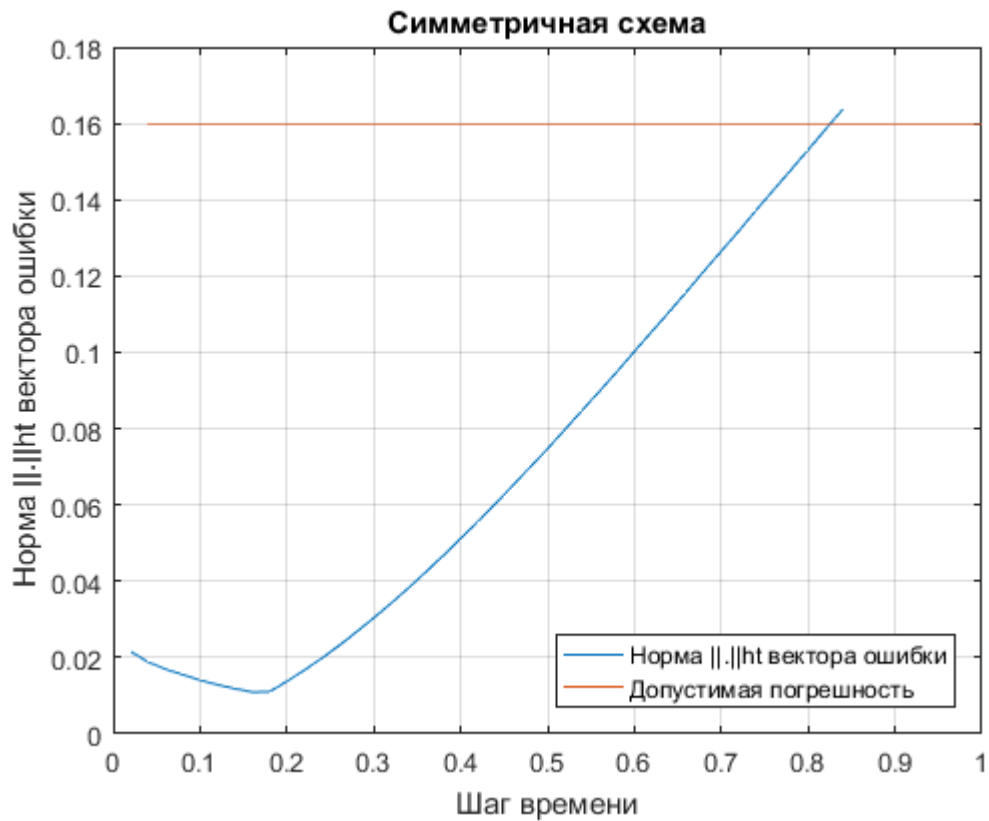


Рис. 4 Нахождение максимального шага по времени для симметричной схемы

В симметричной схеме мы смогли сделать шаг по времени гораздо больше, чем в неявной схеме $\tau = 0.8$

Сравнение эффективности схем

Будем сравнивать разностные схемы по времени работы программы.

Построим графики зависимости работы программы от точности решения. Шаг по времени для каждой схемы возьмем максимально возможный, чтобы ошибка численного решения была не больше ε . Тогда для явной схемы это $\tau_0 = 5 \cdot 10^{-4}$, для неявной схемы - $\tau = 0.2$, а для явной схемы - $\tau = 0.8$

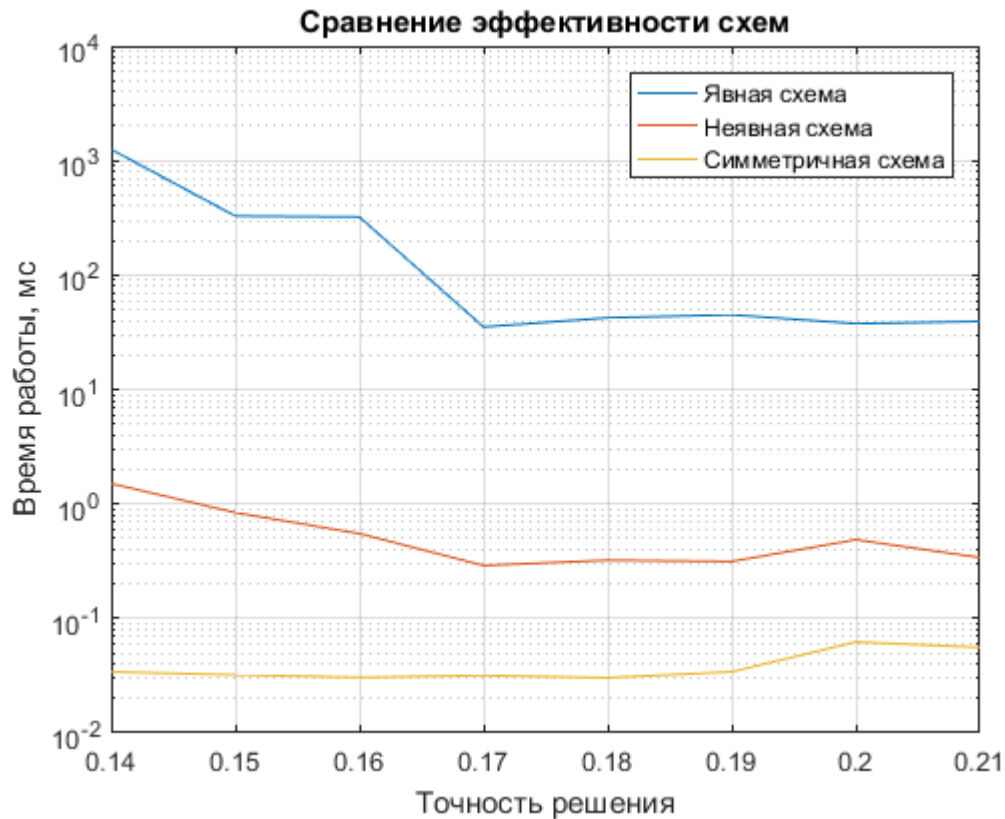


Рис. 5 Сравнение эффективности схем

Симметричная схема оказалась наиболее эффективной, так как находит решение с нужной точностью за наименьшее время. Явная схема оказалась наименее эффективной. Это связано с тем, что для достижения заданной точности в явной схеме необходимо брать очень маленькие шаги по времени, вследствие чего сильно возрастает число итераций алгоритма и время его работы. Аналогично, симметричная схема требует наименьшее время, так как для достижения заданной точности она требует шаг по времени, гораздо больший чем в явной или неявной схемах.

Графики точного и численных решений

Построим два графика точной и численных функций решения параболического уравнения, которые вышли на устойчивое решение.

Сначала положим $x = \frac{1}{2}$ и построим график зависимости точного и численного решения в зависимости от времени.

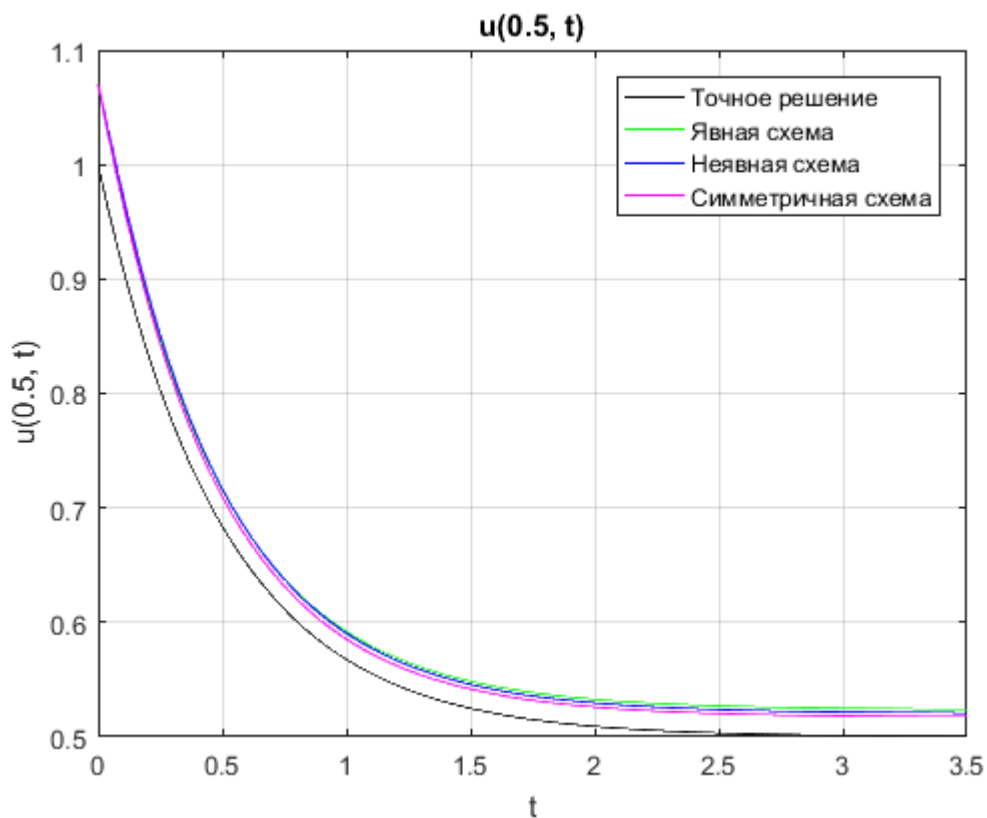


Рис. 6 Графики функций $u(1/2, t)$ точного и численных решений

Из графика можно сделать вывод, что все три схемы нашли решения, которые сходятся к нужному установившемуся решению с хорошей точностью.

Теперь положим $t = 2$ и построим график зависимости точного и численного решения в зависимости от координаты.

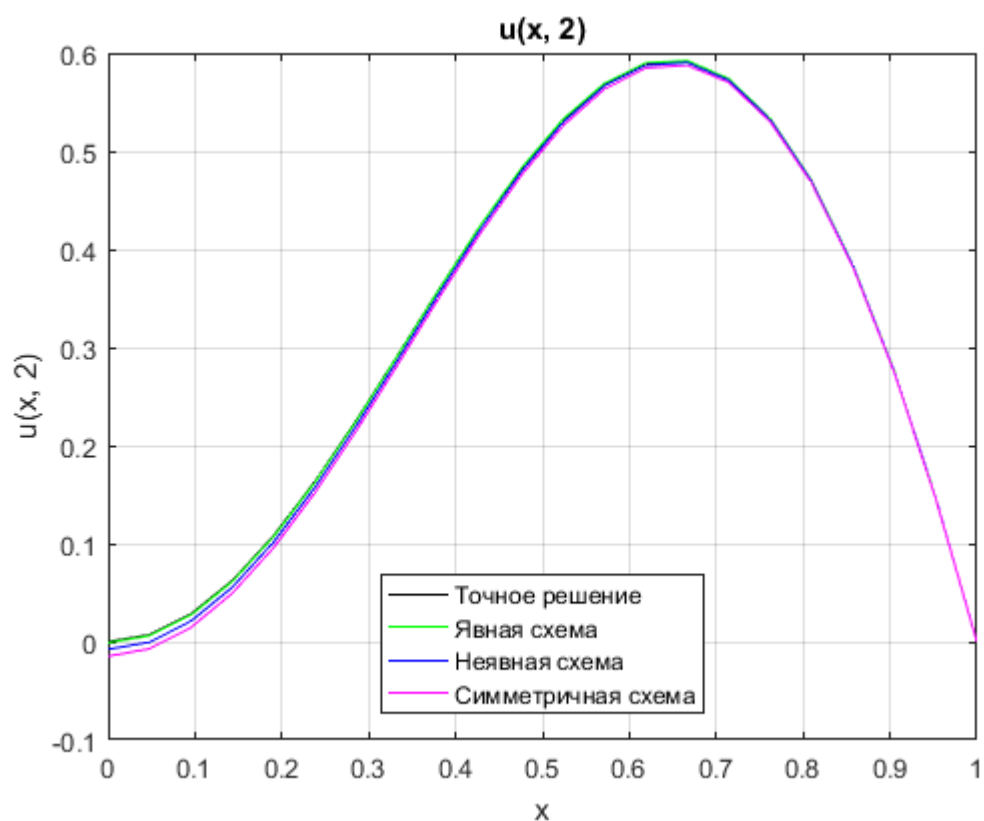


Рис. 7 Графики функций $u(x, 2)$ точного и численных решений

Из графика видно, что для большого t все три схемы дали решения, которые хорошо точно описывают решение во всех точках пространства.

Вывод

Все три схемы (явная, неявная, симметричная) гарантированно находят решения при соблюдении условий сходимости. Явная схема гораздо более проста в реализации и в вычислительной сложности, чем неявная и симметричная схемы. Однако у явной схемы более жесткие условия на сходимость, а именно шаг по времени должен быть достаточно мал. Неявная и симметричная схемы абсолютно устойчивы и не имеют таких жестких требований на сходимость, вследствие чего они показали себя более эффективными по времени работы алгоритма, так как можно взять большой шаг по времени и тем самым сократить число итераций алгоритма.