то для сходимости $\left\|x - x_{n,\delta}\right\|_A \to 0, n \to \infty$, достаточно, чтобы $n^{\frac{1}{4}}\delta \to 0, n \to \infty, \delta \to 0$.

Итак, доказана

Теорема 2. При условии $\alpha > 0$ итерационный метод (3) сходится в энергетической норме гильбертова пространства, если число итераций *n* выбирать из условия $n^{\frac{1}{4}}\delta \to 0$ при $n \to \infty, \delta \to 0$.

Запишем теперь общую оценку погрешности для метода (3) в энергетической норме

$$\left\|x - x_{n,\delta}\right\|_A \le (4n\alpha e)^{-\frac{1}{4}}\left\|x\right\| + 2^{\frac{3}{2}}(n\alpha)^{\frac{1}{4}}\delta, n \ge 1. \qquad (5)$$

Оптимизируем оценку (5) по n. Для этого при заданном δ найдем такое значение числа итераций *n,* при котором оценка погрешности становится минимальной. Приравняв нулю производную по n от правой части неравенства (5), получим

$$n_{опт} = 2^{-1}\alpha^{-1}e^{-\frac{1}{2}}\delta^{-2}\left\|x\right\|^2. \qquad (6)$$

Подставив $n_{опт}$ в оценку (5), найдем её оптимальное значение

$$\left\|x - x_{n,\delta}\right\|_A^{опт} \le 2^{\frac{3}{2}}e^{-\frac{1}{8}}\delta^{\frac{1}{2}}\left\|x\right\|^{\frac{1}{2}}. \qquad (7)$$

Таким образом, справедлива.

Теорема 3. Оптимальная оценка погрешности для метода (3) при условии $\alpha > 0$ в энергетической норме имеет вид (7) и получается при $n_{опт}$ из (6).

Замечание 2. Из неравенства (7) вытекает, что оптимальная оценка погрешности не зависит от параметра α. Но $n_{опт}$ зависит от α и, поскольку на α нет ограничений сверху ($\alpha > 0$), то за счет выбора α можно получить $n_{опт} = 1$, то есть оптимальная оценка погрешности будет достигаться уже на первом шаге итераций. Для этого достаточно взять $\alpha_{опт} = 2^{-1}e^{-\frac{1}{2}}\delta^{-2}\left\|x\right\|^2$.

Рассмотрим вопрос о том, когда из сходимости в энергетической норме следует сходимость в обычной норме гильбертова пространства *H*. Очевидно, для этого достаточно, чтобы при некотором фиксированном $\varepsilon\ (0<\varepsilon<\|A\|)$, было $P_\varepsilon x = 0, P_\varepsilon x_{n,\delta} = 0$, где $P_\varepsilon = \int_0^\varepsilon \lambda dE_\lambda$.

Так как $x_{n,\delta} = A^{-1}\left[E - \left(E + \alpha A^2\right)^{-n}\left(E - \alpha A^2\right)^n\right]y_\delta$, то для выполнения последнего из указанных условий должно выполняться условие $P_\varepsilon y_\delta = 0$. Таким образом, если решение x и приближенная правая часть y_δ таковы, что $P_\varepsilon x = 0$ и $P_\varepsilon y_\delta = 0$, то из сходимости $x_{n,\delta}$ к x в энергетической норме вытекает сходимость в исходной норме гильбертова пространства H, и, следовательно, для сходимости в исходной норме пространства H не требуется истокопредставимости точного решения.

Для решения уравнений с несамосопряженным или неположительным, но ограниченным оператором A следует перейти к уравнению $A^*Ax = A^*y$. Тогда при приближенном элементе y_δ метод (3) примет вид

$$\left(E + \alpha\left(A^*A\right)^2\right)x_{n+1\delta} = \left(E - \alpha\left(A^*A\right)^2\right)x_{n,\delta} + 2\alpha A^{*2}Ay_\delta,\ x_{0,\delta} = 0.$$

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Bialy, H. Iterative Behandlung linearer Funktionsgleichungen / H. Bialy // Arch. Ration. Mech. and Anal. – 1959. – Vol. 4, № 2. – P. 166–176.
2. Савчук, В.Ф. Об одном неявном итеративном методе решения операторных уравнений / В.Ф. Савчук, О.В. Матысик // Доклады НАН Беларуси. – 2005. – Т. 49. – № 4. – С. 38–42.
3. Матысик, О.В. Сходимость в гильбертовом пространстве неявной итерацион-ной процедуры решения линейных уравнений / О.В. Матысик, В.Ф. Савчук // Вестник Брестского университета. – 2008. – № 1(30). – С. 15–21.
4. Лисковец, О.А. Сходимость в энергетической норме итеративного метода для уравнений I-го рода / О.А. Лисковец, В.Ф. Савчук // Известия АН БССР. Сер. физ.-матем. наук. – 1976. – № 2. – С.19–23.
5. Савчук, В. Ф. Неявная итерационная процедура решения операторных уравнений в гильбертовом пространстве / В.Ф. Савчук, О.В. Матысик // Доклады НАН Беларуси. – 2006. – Т. 50. – № 5. – С. 37–42.

Материал поступил в редакцию 12.09.2017

MATYSIK O.V., SIDAK S.V. Regularization of ill-posed problems in the case of non-uniqueness of the solution of operator equations of the first kind

In the Hilbert space for solving operator equations of type I with affirmative limited and self–conjugate operator the implicite iteration method is proposed. The case of non–uniqueness of solving operator equation is investigated. It is shown, that in this case the iteration method converges to the decision with the minimal norm. In energy norm of Hilbert space for the proposed method convergence is proved and apriori estimations of this method error have been received. Use of energy norm allows to make a method quite effective even then when there are no data about source representability of exact solution of the equation.

УДК 519.6 + 517.983.54

Матысик О.В., Сидак С.В.

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ НАХОЖДЕНИЯ ПРИБЛИЖЕННОГО РЕШЕНИЯ МОДЕЛЬНОЙ НЕКОРРЕКТНОЙ ЗАДАЧИ С ПОМОЩЬЮ НЕЯВНОЙ ИТЕРАЦИОННОЙ ПРОЦЕДУРЫ В ГИЛЬБЕРТОВОМ ПРОСТРАНСТВЕ

Введение. В гильбертовом пространстве для решения операторного уравнения первого рода с положительным ограниченным и самосопряженным оператором изучается неявный итерационный метод. Для предложенного метода обосновано применение правила останова по невязке, что делает рассматриваемый итерационный метод эффективным и тогда, когда нет сведений об истокообразной представимости точного решения. Рассматриваемым методом решена численная модельная некорректная задача в виде интегрального уравнения Фредгольма первого рода.

1. Постановка задачи. В действительном гильбертовом пространстве H исследуется операторное уравнение I рода

$$Ax = y\,, \tag{1}$$

где A – положительный ограниченный и самосопряженный оператор, для которого нуль не является собственным значением, однако принадлежит спектру оператора A, и, следовательно, задача некорректна. Пусть $y \in R(A)$, т. е. при точной правой части y уравнение (1) имеет единственное решение x. Для отыскания этого решения применяется неявная итерационная процедура

$$\left(E + \alpha A^2\right) x_{n+1} = \left(E - \alpha A^2\right) x_n + 2\alpha A y, \quad x_0 = 0\,. \tag{2}$$

В случае приближенной правой части y_δ, $\left(\|y - y_\delta\| \le \delta\right)$ соответствующие методу (2) итерации примут вид

$$\left(E + \alpha A^2\right) x_{n+1,\delta} = \left(E - \alpha A^2\right) x_{n,\delta} + 2\alpha A y_\delta, \quad x_{0,\delta} = 0\,. \tag{3}$$

Ниже, как обычно, под сходимостью метода (3) понимается утверждение о том, что приближения (3) сколь угодно близко подходят к точному решению уравнения при достаточно малых δ и $n\delta$ и достаточно больших n.

2. Сходимость метода в случае априорного выбора числа итераций. Методы (2)–(3) были предложены в работе [1], в которой показана сходимость обоих методов (2) и (3) при условии $\alpha > 0$ и получена оценка погрешности в предположении, что решение является истокообразно представимым с некоторым показателем $s > 0$. В статье [1] доказана

Теорема 1. *Если* $x = A^s z,\ s > 0$, *и* $\alpha > 0$, *то справедлива оценка погрешности для метода итераций (3):*

$$\|x - x_{n,\delta}\| \le s^{s/2} (2n\alpha e)^{-s/2} \|z\| + 4n^{1/2}\alpha^{1/2}\delta,\ \ n \ge 1$$

Отсюда при

$$n_{опт} = 2^{-2/(s+1)} \left(\frac{s}{2}\right)^{(s+2)/(s+1)} \alpha^{-1} e^{-s/(s+1)} \|z\|^{2/(s+1)} \delta^{-2/(s+1)}$$

получаем неравенство

$$\|x - x_{n,\delta}\|_{опт} \le (1+s)\cdot 2^{s/(s+1)} \left(\frac{s}{2}\right)^{s(1-2)/(2(s+1))} e^{-s/(2(s+1))} \delta^{s/(s+1)} \|z\|^{1/(s+1)}.$$

Порядок последней оценки оптимален в классе задач с истокообразно представимыми решениями [2].

Очевидно, что оптимальная оценка погрешности не зависит от параметра α, но от него зависит $n_{опт}$. Так как на α нет ограничений сверху ($\alpha > 0$), то можно α выбрать так, чтобы $n_{опт} = 1$. Для этого достаточно взять

$$\alpha_{опт} = 2^{-2/(s+1)} \left(\frac{s}{2}\right)^{(s+2)/(s+1)} e^{-s/(s+1)} \|z\|^{2/(s+1)} \delta^{-2/(s+1)}.$$

Таким образом, неявный метод (3) позволяет получить решение уже на первых шагах итераций.

3. Правило останова по невязке. Для решения операторного уравнения (1) с ограниченным положительным и самосопряжённым оператором применяется метод итераций (3). Покажем возможность применения для этого метода правила останова по невязке [2–4].

Определим момент m останова итерационного процесса (3) условием

$$\|Ax_{n,\delta} - y_\delta\| > \varepsilon,\ (n < m)\,, \tag{4}$$

$$\|Ax_{m,\delta} - y_\delta\| \le \varepsilon, \quad \varepsilon = b\delta,\ \ b > 1.$$

Предполагается, что при начальном приближении $x_{0,\delta}$ невязка достаточно велика, больше уровня останова ε, т. е. $\|Ax_{0,\delta} - y_\delta\| > \varepsilon$. Справедливы [3]

Т е о р е м а 2. *Пусть* $A = A^* \ge 0,\ \|A\| \le M$ *и пусть момент останова* $m = m(\delta)$ *в методе (3) выбирается по правилу (4). Тогда* $x_{m,\delta} \to x$ *при* $\delta \to 0$.

Т е о р е м а 3. *Пусть выполнены условия теоремы 2 и пусть* $x = A^s z, s > 0$. *Тогда справедливы оценки* $m \le 1 + \frac{s+1}{2\alpha e}\left[\frac{\|z\|}{(b-1)\delta}\right]^{\frac{2}{s+1}}$,

$$\|x_{m,\delta} - x\| \le \left[(b+1)\delta\right]^{\frac{s}{s+1}} \|z\|^{\frac{1}{s+1}} + 4\alpha^{\frac{1}{2}} \left\{1 + \frac{s+1}{2\alpha e}\left[\frac{\|z\|}{(b-1)\delta}\right]^{\frac{2}{s+1}}\right\}^{\frac{1}{2}} \delta. \tag{5}$$

З а м е ч а н и е 1. *Порядок оценки (5) есть* $O\left(\delta^{s/(s+1)}\right)$ *и, как следует из [2], он оптимален в классе задач с истокопредставимыми решениями.*

З а м е ч а н и е 2. *Используемое в формулировке теоремы 3 предположение порядка* $s > 0$ *истокопредставимости точного решения не потребуется на практике, так как оно не содержится в правиле останова (4).*

4. Численный модельный пример

4.1. Формулировка и описание алгоритма решения модельной задачи

Задача. Решаем в пространстве $L_2(0,1)$ модельную задачу в виде уравнения

$$\int_0^1 K(t,s)\, x(s)\, ds = y(t), \quad 0 \le t \le 1 \tag{6}$$

с симметричным положительным ядром

$K(t,s) = \begin{cases} t(1-s),\ 0 \le t \le s \le 1, \\ s(1-t),\ 0 \le s \le t \le 1, \end{cases}$ точной правой частью

$y(t) = \dfrac{t(t-1)(t^2-t-1)}{12}$ и точным решением $x(t) = t(1-t)$

Обычно на практике мы не знаем точной функции $y(t)$, а вместо нее известны значения приближенной функции $\tilde{y}(t)$ в некотором числе точек с определенной, часто известной погрешностью δ, и по этим приближенным данным требуется приближенно найти решение. Чтобы имитировать эту ситуацию, будем считать заданными значения $\tilde{y}_i$, $i = \overline{1,m}$, полученные следующим образом: $\tilde{y}_i = [\,y(t_i)\cdot 10^k + 0{,}5] / 10^k$, где $y(t_i)$ - значения функции $y(t)$ в точках $t_i = ih$, $i = \overline{1,m}$, $h = 1/m$. Квадратные скобки означают целую часть числа и $k = 4$. При $k = 4$ величина погрешности $\delta = 10^{-4}$. Действительно, имеем $\int_0^1 \left[y(t) - \tilde{y}(t)\right]^2 dt \approx$

$\approx \sum_{i=1}^m \left[y(t_i) - \tilde{y}_i\right]^2 h \le mh\left(10^{-k}\right)^2 = 10^{-2k}$. Заменим интеграл в уравнении (6) квадратурной суммой, например, по формуле правых прямоугольников с узлами $s_j = jh$, $j = \overline{1,m}$ $h = 1/m$, т. е.

$\int_0^1 K(t,s)\, x(s)\, ds \approx \sum_{j=1}^m K(t,s_j) h x_j$. Тогда получим равенство

$\sum_{j=1}^m K(t,s_j) h x_j + \rho_m(t) = y(t)$, где ρ_m - остаток квадратурной замены. Записав последнее равенство в точках $t_i = ih$, $i = \overline{1,m}$, получим уравнения $\sum_{j=1}^{m} K\left(t_i, s_j\right) h x_j + \rho_m\left(t_i\right) = y\left(t_i\right)$, $i = \overline{1,m}$.

Отбросив теперь остаточный член, получим линейную алгебраическую систему уравнений относительно приближённого решения

$$\sum_{j=1}^{m} K\left(t_i, s_j\right) h x_j = \tilde{y}_i, \; i = \overline{1,m}. \tag{7}$$

Выберем для определённости m=32 и будем решать систему (7) методом итераций (3), который в дискретной форме запишется

$$x_i^{(n+1)} + \alpha \sum_{j=1}^{m} K\left(t_i, s_j\right) h \left(\sum_{k=1}^{m} K\left(t_j, s_k\right) h x_k^{(n+1)} \right) = x_i^{(n)} -$$

$$-\alpha \sum_{j=1}^{m} K\left(t_i, s_j\right) h \left(\sum_{k=1}^{m} K\left(t_j, s_k\right) h x_k^{(n)} \right) + 2\alpha \tilde{y}_i, \; i = \overline{1,m}. \tag{8}$$

При решении задачи неявным итерационным методом (3) вычислялись:

$\left\| A x^{(n)} - \tilde{y} \right\|_m = \left\{ \sum_{i=1}^{m} \left[\sum_{j=1}^{m} K\left(t_i, s_j\right) h x_j^{(n)} - \tilde{y}_i \right]^2 h \right\}^{1/2}$ - дискретная норма невязки, $\left\| x^{(n)} \right\|_m = \left\{ \sum_{i=1}^{m} \left[x_i^{(n)} \right]^2 h \right\}^{1/2}$ - норма приближённого решения и дискретная норма разности между точным и приближённым решениями: $\left\| x - x^{(n)} \right\|_m = \left\{ \sum_{i=1}^{m} \left[x\left(t_i\right) - x_i^{(n)} \right]^2 h \right\}^{1/2}$.

Оператор, описанный выше интегральным уравнением, непрерывен, взаимно однозначен и аддитивен. Задача была решена методом (3) при $\delta = 10^{-4}$. Результаты счёта приведены в таблице 1 (ввиду симметрии приведена лишь половина таблицы). Для решения предложенной задачи сведений об истокопредставимости точного решения не потребовалось, так как здесь воспользовались правилом останова по невязке (4), выбрав уровень останова $\varepsilon = 1{,}5\delta$. Пример расчёта показал, что для достижения оптимальной точности методом (3) при $\alpha = 9$ требуется только одна итерация, что соответствует результатам раздела 2. На рисунке 1 изображены графики точного решения и приближённого решения, полученного методом (3) при $\delta = 10^{-4}$.

4.2 Результат работы программы

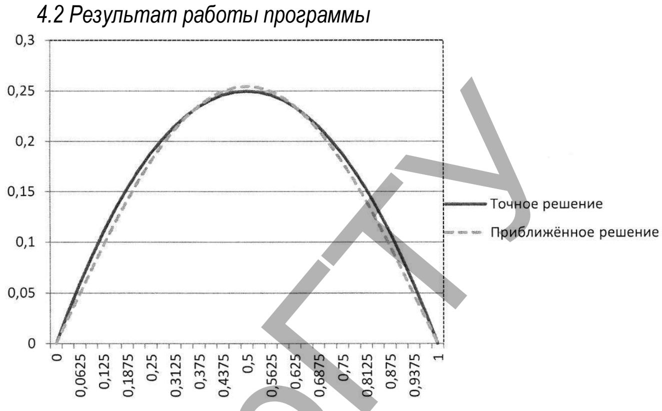

Рисунок 1

4.3. Исходный код программы (на языке программирования C#)

<u>Вспомогательные классы Matrix и Vector</u>:

```
using System;
using MTask.CoreClass;
namespace Helpers.Classes
{
 public class Matrix
 {
 private int _height, _width;
 private double[,] data;
 private double[] t;
 public int Height { get { return _height; } }
 public int Width { get { return _width; } }
 public double[,] Data { get { return data; } set { data = value; } }
}
 public Matrix(int height, int width)
 {
```

Таблица 1

Узлы t_i	Правые части $y\left(t_i\right)$	Точное решение $x\left(t_i\right)$	Приближённое решение, полученное методом (3) $\delta = 10^{-4}$
0	**0**	**0**	**0**
0.0312	**0.00259**	**0.03027**	**0.02429**
0.0625	**0.00517**	**0.05859**	**0.04865**
0.0937	**0.00768**	**0.08496**	**0.07275**
0.125	**0.01011**	**0.10938**	**0.09629**
0.1562	**0.01243**	**0.13184**	**0.11898**
0.1875	**0.01463**	**0.15234**	**0.14056**
0.2187	**0.01668**	**0.17089**	**0.1608**
0.2500	**0.01855**	**0.1875**	**0.17948**
0.2812	**0.02025**	**0.20215**	**0.19641**
0.3125	**0.02175**	**0.21484**	**0.21142**
0.3437	**0.02304**	**0.22559**	**0.22437**
0.375	**0.02411**	**0.23438**	**0.23514**
0.4062	**0.02495**	**0.24121**	**0.24361**
0.4375	**0.02555**	**0.24609**	**0.24972**
0.4687	**0.02591**	**0.24902**	**0.25341**
0.5000	**0.02604**	**0.25000**	**0.25464**
$\left\| A x^{(n)} - \tilde{y} \right\|_m$			**0.00015**
$\left\| x^{(n)} \right\|_m$			**0.17972**
$\left\| x - x^{(n)} \right\|_m$			**0.00798**

```
_height = height; _width = width;
data = new double[_height, _width];
}
private int getFirstNonZero(int RowNum)
{
for (int i = 0; i < _width; i++)
if (Data[RowNum, i] != 0) return i;
return _width;
}
private void Sort(Vector temp)
{
t = new double[_width];
for (int i = 0; i < _height - 1; i++)
for (int j = i + 1; j < _height; j++)
if (getFirstNonZero(i) > getFirstNonZero(j))
{
for (int h = 0; h < _width; h++) t[h] = Data[i, h];
for (int h = 0; h < _width; h++) Data[i, h] = Data[j, h];
for (int h = 0; h < _width; h++) Data[j, h] = t[h];
double t1 = temp.Data[i];
temp.Data[i] = temp.Data[j];
temp.Data[j] = t1;
}
}
private void ToStage(Vector temp)
{
double alpha;
for (int i = 0; i < _height; i++)
{
if (Data[i, i] == 0) continue;
for (int j = i + 1; j < _height; j++)
{
if (Data[j, i] == 0) continue;
alpha = Data[j, i] / Data[i, i];
for (int h = 0; h < _width; h++)
Data[j, h] -= Data[i, h] * alpha;
temp.Data[j] -= temp.Data[i] * alpha;
Data[j, i] = 0;
}
Sort(temp);
}
Sort(temp);
}
public void CopyFrom(Matrix m)
{
Data = new double[m._height, m._width];
_height = m._height; _width = m._width;
for (int i = 0; i < m._height; i++)
for (int j = 0; j < m._width; j++)
Data[i, j] = m.Data[i, j];
}
public Vector GetSolution(Vector b)
{
Vector res = new Vector();
Vector temp = new Vector();
temp.CopyFrom(b);
ToStage(temp);
if (Data[_height - 1, _width - 1] == 0)
res.Data[_width - 1] = 0;
else
res.Data[_width - 1] = temp.Data[_width - 1] /
Data[_height - 1, _width - 1];
for (int i = _height - 2; i >= 0; i--)
{
if (Data[i, i] == 0)
res.Data[i] = 0; continue;
double sum = 0;
for (int j = i + 1; j < _width; j++)
sum += Data[i, j] * res.Data[j];
temp.Data[i] -= sum;
res.Data[i] = temp.Data[i] / Data[i, i];
}
return res;
}
}
public class Vector
{
private int _dimension;
private double[] data;
public int Dimension { get { return _dimension; } }
public double[] Data { get { return data; } set { data = value; } }
public Vector(int dimension)
{
_dimension = dimension;
data = new double[dimension];
}
public Vector()
{
_dimension = Core.M;
data = new double[Core.M];
}
public void CopyFrom(Vector w)
{
this._dimension = w._dimension;
for (int i = 0; i < _dimension; i++)
this.Data[i] = w.Data[i];
}
}
}
```

<u>Вспомогательный класс Core</u> (решение задачи):

```
using System;
using Helpers.Classes;
namespace MTask.CoreClass
{
public class Core
{
private double h;
private Matrix K, Z;
private double n1, n2, n3;
private int iterations;
private Vector solution, v1, v2, y, x0;
public const int M = 32, k = 4;
public const double alpha = 9, a = 0, b = 1, delta = 1e-4, eps =
1.5 * delta;
public double Norm1 { get { return n1; } }
public double Norm2 { get { return n2; } }
public double Norm3 { get { return n3; } }
public Vector PreciselySolution { get { return v2; } }
public Vector Solution { get { return solution; } }
public int TotalIterations { get { return iterations; } }
public Core()
{
v1 = new Vector(); v2 = new Vector(); x0 = new Vector();
y = new Vector(); solution = new Vector();
K = new Matrix(M, M);
}
private void GetMatrix()
{
for (int i = 0; i < M; i++) v1.Data[i] = i * h;
for (int i = 0; i < M; i++)
for (int j = 0; j < M; j++)
K.Data[i, j] = GetK(v1.Data[i], v1.Data[j]);
```

```
}
private double GetPoint(int i)
{
return i * h;
}
public double GetY(int i)
{
double f = GetPoint(i);
return f * (f - 1) * (f * f - f - 1) / 12;
}
private double GetK(double t, double s)
{
if (t <= s) return t * (1 - s);
if (s <= t) return s * (1 - t);
return 0;
}
private Vector GetRightPart(Vector xn)
{
Vector res = new Vector();
for (int k = 0; k < M; k++)
{
double t1 = GetY(k);
double t2 = 0;
for (int j = 0; j < M; j++) t2 += K.Data[k, j] * xn.Data[j] * h;
res.Data[k] = xn.Data[k] - alpha * t2 + 2 * alpha * t1;
}
return res;
}
public double GetQ(Vector v)
{
double sum = 0;
for (int i = 0; i < M; i++)
{
double t = 0;
for (int j = 0; j < M; j++)
t += K.Data[i, j] * h * v.Data[j];
t -= (GetY(i) * Math.Pow(10, k) + 0.5) / Math.Pow(10, k);
t *= t;
t *= h;
sum += t;
}
return Math.Sqrt(sum);
}
public void Solve()
{
h = (b - a) / M;
iterations = 0;
x0 = new Vector();
solution = new Vector();
GetMatrix();
Z = new Matrix(M, M);
for (int i = 0; i < M; i++)
{
double t0 = 1 + alpha * K.Data[i, 0] * h;
Z.Data[i, i] = t0;
for (int j = 0; j < M; j++)
{
if (i == j) continue;
Z.Data[i, j] = alpha * K.Data[i, j] * h;
}
}
Vector bb = GetRightPart(x0);
solution = Z.GetSolution(bb);
n1 = GetQ(solution);
for (int i = 0; i < M; i++)
 v2.Data[i] = GetPoint(i) * (1 - GetPoint(i));
 double t = 0;
 for (int i = 0; i < M; i++)
 t += solution.Data[i] * solution.Data[i] * h;
 n2 = Math.Sqrt(t); t = 0;
 for (int i = 0; i < M; i++)
 t += (v2.Data[i] - solution.Data[i]) *
 (v2.Data[i] - solution.Data[i]) * h;
 n3 = Math.Sqrt(t);
 }
 }
}
```

<u>*Главный класс программы*</u>*:*

```
using System;
using MTask.CoreClass;
namespace MTask
{
 class Program
 {
 static void Main(string[] args)
 {
 Core core = new Core();
 core.Solve();
 Console.WriteLine("Precisely solution: ");
 for (int i = 0; i < 32; i++)
 Console.WriteLine("x" + i + ": " +
core.PreciselySolution.Data[i]);
 Console.WriteLine();
 Console.WriteLine("Approximately solution: ");
 for (int i = 0; i < 32; i++)
 Console.WriteLine("x" + i + ": " + core.Solution.Data[i]);
 Console.WriteLine();
 Console.WriteLine("Total iterations: ");
 Console.WriteLine(core.TotalIterations);
 Console.WriteLine();
 Console.WriteLine("Norm: ");
 Console.WriteLine(core.Norm1);
 Console.WriteLine(core.Norm2);
 Console.WriteLine(core.Norm3);
 Console.WriteLine();
 Console.WriteLine("Ti");
 for (int i = 0; i < 17; i++)
 Console.WriteLine((Core.b - Core.a) / Core.M * (i));
 Console.WriteLine();
 Console.WriteLine("Yi");
 for (int i = 0; i < 17; i++)
 Console.WriteLine(core.GetY(i));
 }
 }
}
```

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Матысик, О.В. Сходимость в гильбертовом пространстве неявного итерационного процесса решения некорректных уравнений первого рода / О.В. Матысик, С.В. Сидак // Весн. Брэсц. ун-та. – Сер. 4: Фізіка. Матэматыка. – 2016. – № 2. – С. 71–76.
2. Вайникко, Г.М. Итерационные процедуры в некорректных задачах / Г.М. Вайникко, А.Ю. Веретенников. – М.: Наука, 1986. – 178 с.
3. Матысик, О.В. Явные и неявные итерационные процедуры решения некорректно поставленных задач / О.В. Матысик. – Брест: Брест. гос. ун-т, 2014. – 213 с.
4. Матысик, О.В. Итерационная регуляризация некорректных задач / О.В. Матысик. – Saarbrücken: LAP LAMBERT Academic Publishing, 2015. – 188 с.

Материал поступил в редакцию 04.02.2017

MATYSIK O.V., SIDAK S.V. Software implementation finding approximate solution of the model ill-posed problem by implicit iteration procedure in Hilbert space

In the Hilbert space for solving operator equations of type I with affirmative limited and self–conjugate operator the implicite iteration method is studies. The application of a rule residual stop for the offered method has been proved, which makes viewed iteration method quite effective even then when there are no data about source representability of exact solution. The viewed method solves numerical model incorrect task as integral equation Fredgolm's of one sort.

УДК 004.8.032.2

Глущенко Т.А., Касьяник В.В., Пролиско Е.Е., Шуть В.Н.

ИНФОБУС – НОВЫЙ ТИП ИНТЕЛЛЕКТУАЛЬНОГО ТРАНСПОРТА ДЛЯ ВНУТРИГОРОДСКИХ ПАССАЖИРСКИХ ПЕРЕВОЗОК

Введение. Автомобили без водителя – транспорт будущего, который уже сейчас постепенно входит в нашу жизнь. Все чаще появляются новости о том, что используются транспортные средства без управления человеком. Вместе с развитием таких транспортных средств резко встает вопрос о регулировании движения таких транспортных средств. Данную проблему может решить использование интеллектуальных систем управления движением транспортных средств и интеллектуальных транспортных средств без управления человеком.

Одним из направлений развития интеллектуального транспорта является разработка транспортных средств для внутригородской и междугородней перевозки пассажиров. Важной отличительной характеристикой таких транспортных средств является взаимодействие пассажира с транспортным средством и отсутствие лица, управляющего транспортным средством.

Дальнейшее развитие по этому направлению позволит создать интеллектуальную информационную систему движения транспортных средств, а также разработать новые перспективные системы передвижения пассажиров в улично-дорожной сети города (УДС).

1. Основные проблемы движения в городском трафике. Более конкретно проблема современной транспортной системы заключается в методах, которые используются для обеспечения должного уровня её работы. Перечислим эти проблемы: проблема неоптимизированного роста, проблема децентрализации и проблема низкой адаптивности.

Проблема неоптимизированного роста заключается в том, что для увеличения объёма перевозимых ресурсов (в том числе пассажиров) производится и внедряется большее количество транспортных единиц (или машин), но не оптимизируется их передвижение. Это, в свою очередь, приводит к ряду других проблем.

Во-первых, возрастание количества машин на отдельном участке или неизбежно влечёт за собой затруднение движения не только на данном, но и на других, связанных с ним, участках транспортной сети.

Во-вторых, подобный рост не выгоден экономически. Помимо необходимости производства дополнительного средства передвижения, повышается потребность в найме обслуживающего персонала – водителей и механиков.

В-третьих, такой путь развития вреден для окружающей среды, учитывая, что в настоящий момент используются преимущественно двигатели внутреннего сгорания.

В-четвёртых, это приводит к возрастанию сложности регулирования транспортных потоков, что формирует проблему децентрализации.

Проблема децентрализации заключается в отсутствии взаимосвязи между большей частью участников движения. Даже косвенная связь посредством расписания движения, что реализовано для автобусов, не может заменить ни коммуникации между участникам, ни работы системы центрального управления. Причины этого следующие.

Во-первых, каждый участник движения может только предполагать намерения других участников, но не может знать их наверняка. Результатом этого являются ДТП и вынужденные задержки при передвижении.

Во-вторых, имеет место неравномерное распределение транспортных потоков по системе: для их систематизации используются, например, полосы или улицы с направленным движением, что может привести к перегрузке одних дорожных участков и низкой загруженности других. Таким образом, дополнительные ресурсы, выделенные для работы системы, в определённое время оказываются избыточными и не реализуются.

В-третьих, система включает в себя как общественный, так и личный транспорт. Это вынуждает их разделять общий ресурс и, тем самым, ограничивает степень реализации обоих. Кроме того, это делает невозможным покрытие всей совокупности пунктов назначения сетью общественного транспорта.

В совокупности эти проблемы вынуждают граждан к очень активному использованию общественного транспорта, но лишь в отдельные и довольно короткие промежутки времени. Иными словами, личный транспорт часто не задействуется большую часть времени, являя собой практически нереализуемый ресурс, что критично в условиях нехватки последних. Из этого напрямую следует проблема низкой адаптивности.

Проблема низкой адаптивности заключается в практически полном отсутствии зависимости количества используемых средств передвижения от количества людей, которым необходимо в данный момент ими воспользоваться.

Проявления этой проблемы следующие. Во-первых, даже в те моменты, когда используется личный автотранспорт, он задействуется для перевозки небольшого количества пассажиров (около полутора человек на единицу), создавая нагрузку на сеть немногим меньшую, чем средства общественного транспорта.

Во-вторых, в час-пик стабильно наблюдается перегрузка средств общественного транспорта. Имеется в виду как физическая его перегрузка, так и неспособность обеспечить необходимую пропускную способность пассажиров.

В-третьих, экономическая составляющая этой инфраструктуры не позволяет применить экстенсивные методы, как в случае с общим ростом объёмов перевозок: как правило, даже без дополнительного увеличения количества машин в то время, когда нагрузка меньше критической, коэффициент их полезного действия существенно падает в сравнении с этим же коэффициентом во время часа пик. Под

Глущенко Татьяна Александровна, *старший преподаватель кафедры «Интеллектуальные информационные технологии» Брестского государственного технического университета.*

Касьяник Валерий Викторович, *старший преподаватель кафедры «Интеллектуальные информационные технологии» Брестского государственного технического университета.*

Пролиско Евгений Евгеньевич, *к.т.н., доцент кафедры «Интеллектуальные информационные технологии» Брестского государственного технического университета.*

Шуть Василий Николаевич, *к.т.н., доцент кафедры «Интеллектуальные информационные технологии» Брестского государственного технического университета.*

Беларусь, БрГТУ, 224017, г. Брест, ул. Московская, 267.