УДК 004.051

I. Shakenov

Al-Farabi Kazakh National University, Republic of Kazakhstan, Almaty
E-mail: ilias.shakenov@gmail.com

Comparing different degrees of nonlinearity for inverse problem for parabolic equation

In this work we consider one dimensional nonlinear parabolic equation with unknown function on the right side of space variable. As an additional information we are given a function which describes a solution on the left side and thus the problem is overdefined on the left side. The problem is solved by gradient method. The main target is to understand an influence of the nonlinearity degree of the equation on convergence of the numerical algorithm. For that we take different degrees of nonlinear term in the equation, construct a numerical solution and give the results in graphical form. Also we enlarge a time interval and consider a convergence of the algorithm. Some negative effects can be avoided by enlarging the time interval. We give all formulae to solve a direct problem and adjoint problem, give references where to find how to obtain a gradient for the functional given on nonlinear parabolic equation. We also describe the step-by-step algorithm of the solution of the problem. Higher degrees of the nonlinearity make the numerical solution less accurate, but at the same time it makes the functional properties of the equation much better. Influence of these two aspects is considered in the work. Also some comments are given on some moments for the numerical algorithm, such as choosing a constant coefficient in gradient method.
Key words: optimization, control, nonlinear parabolic equation, Gateaux derivative, approximation, gradient.

И. Шакенов

Сравнение разных степеней нелинейности для обратных задач параболического уравнения

В данной работе рассматривается одномерное нелинейное параболическое уравнение с неизвестной функцией на правой границе. В качестве дополнительной информации задано поведение функции на левой границе и тем самым задача переопределена на левой границе. Задача решается градиентным методом. Основная цель – выявить влияние степени нелинейного члена на сходимость численного алгоритма. Для этого выбираются различные степени нелинейности, строится численное решение и результаты представляются в графическом виде. Также рассматривается сходимость алгоритма на более протяженных временных промежутках. Оказывается, некоторое негативные эффекты, влияющие на численное решение, сглаживаются по мере увеличения интервала времени. В работе приводятся расчетные формулы для решения прямой и сопряженной задачи, дается ссылка на статьи, где можно посмотреть вывод градиента функционала для нелинейного параболического уравнения. Приведен также полный пошаговый алгоритм решения задачи. Повышение степени нелинейности уравнения с одной стороны приводит к ухудшению точности численного алгоритма, но с другой стороны улучшает функциональные свойства уравнения. Влияние этих двух эффектов исследуется в данной работе, даются также коментарии и по другим аспектам, вызывающим трудности в численном алгоритме, например, выбор коэффициента при градиенте в градиентном методе.
Ключевые слова: оптимизация, управление, нелинейное параболическое уравнение, производная Гато, приближение, градиент.

И. Шакенов

Параболалық теңдеуі кері есептерінің әртүрлі дәрежелі сызықты еместігін салыстыру

Бұл жұмыста оң жақ шекарасында белгісіз функциясы бар сызықты емес бір өлшемді параболалық теңдеуі қарастырылады. Қосымша мәлімет ретінде сол жақ шекарасында функцияның қасиеті берілген. Сондықтан да есеп сол жақ шекарасында еселі анықталған. Есеп градиент әдісімен шығарылады. Жұмыстың негізгі мақсаты – теңдеудің сызықты емес мүшесінің сандық алгоритмінің жинақталуына әсерін анықтау. Сондықтан да әртүрлі дәрежелі сызықты еместіктері қарастырылады, сандық шешімдері алынады және нәтижелері график түрінде келтіріледі. Сонымен қатар алгоритмнің жинақтылығы үлкен уақыт аралығында қарастырылады. Әйтсе де, сандық шешіміне әсер ететін кейбір қолайсыз эффектілері, уақыт интервалын үлкейту арқылы жақсартылады. Жұмыста тура және түйіндес есептерін шешудің есептеу формулалары келтіріледі, сызықты емес параболалық теңдеуі үшін функционал градиентін есептеуге арналған мақалаларға сілтемелер беріледі. Есепті шешудің толық, қадам сайынғы алгоритмі келтірілген. Сызықты еместігі дәрежесі артқан сайын, бір жағынан, сандық алгоритмнің дәлдігін төмендетеді, ал екінші жағынан, теңдеудің функционалдық қасиетін жақсартады.Осы екі эффектілердің ықпалы осы жұмыста зерттелінеді және де басқа сандық алгоритмінің қыйындылығы туралы жағдайларға пікір айтылады, мысалы, градиент әдісіндегі градиент коэффициентін таңдау.

Түйін сөздер: тиімділік, басқару, сызықты емес параболалық теңдеуі, Гато туындысы, жуықтау, градиент.

Introduction

Problems of optimal control and inverse problems for systems described by linear equations of parabolic type are very well known. [5], [7], [8], [9], [10]. Transition to nonlinear case give rise to considerable difficulties when proving a solvability of optimization problem, deriving conditions of optimality and also numerical methods for their resolution. Due to the impossibility to verify a convexity of the functional, some properties such as uniqueness of the solution, sufficiency of optimality conditions and convergence of iterational methods cannot be investigated. Among the nonlinear problems of the indicated class, problems with exponential nonlinearity have substantial difficulties. Some results in this area one can find in works [6], [11], [12]. But these works are only theoretical. The purpose of this paper is to analyze some algorithmic features of the given optimization problem, in particular, influence of nonlinearity degree and size of time horizon on the effectiveness of numerical algorithm. There were no such researches for indicated type of problems. The least question was considered in author's works for linear systems. [1], [2].

Linear parabolic equation on a fixed time horizon is a very well observed problem even if unknown function is in a boundary condition. We consider a problem for nonlinear parabolic equation in one dimension with time horizons which are small and large as well. Mathematical statement looks as following:

$$\partial_t u(t,x) = \partial_x^2 u(t,x) - u(t,x)|u(t,x)|^p + f(t,x), \quad 0 < t < T, \quad 0 < x < 1, \tag{1}$$

$$u(0,x) = \varphi(x), \quad 0 < x < 1, \tag{2}$$

$$\partial_x u(t,0) = b(t), \quad 0 < t < T, \tag{3}$$

$$\partial_x u(t,1) = y(t), \quad 0 < t < T, \tag{4}$$

Function $y(t)$ is unknown and must be determined. For that purposes we use an additional information $u(t,0) = a(t)$. We transform this problem to optimization problem which requires to minimize a functional

$$I(y) = \int_0^T \big(u(t,0;y) - a(t)\big)^2 dt \to \min$$

If the functional gets its minimum value then $u(t,0)$ is the most close to $a(t)$ and additional information is fulfilled. The most commonly used method for solving such problems is gradient method. For that, we construct a sequence

$$y_{n+1}(t) = y_n(t) - \alpha_n I'\big(y_n(t)\big), \tag{5}$$

where $\alpha_n > 0$. Here the value of $I'\big(y_n(t)\big)$ is given by the following

Теорема 1 *Gateaux derivative of the functional I at the point y is determined by the formula*

$$I'\big(y(t)\big) = \psi(t,1),$$

where $\psi(t,x)$ is the solution of the adjoint problem:

$$\partial_t\psi(t,x) + \partial_x^2\psi(t,x) - (p+1)|u(t,x)|^p\psi(t,x) = 0, \quad 0 < t < T, \quad 0 < x < 1, \tag{6}$$

$$\psi(T,x) = 0, \quad 0 < x < 1, \tag{7}$$

$$\partial_x\psi(t,1) = 0, \quad 0 < t < T, \tag{8}$$

$$\partial_x\psi(t,0) = -2\big(u(t,0;y) - a(t)\big), \quad 0 < t < T, \tag{9}$$

where $0 < t < T$, $0 < x < 1$.

Proofs for linear and nonlinear case of Theorem 1 you can see in [1], [2]. Also see [4], [5] [6], [7], [8] for more different situations.

Algorithm of solving a problem

The common scheme to solve the problem step by step.

1. Initialization of parameters: ε is deviation of $u(t,0;y)$ from $a(t)$ by norm of $\mathbf{H}$; $\tau = \dfrac{T}{M}$ is step by t; $h = \dfrac{1}{N}$ is step by x; set up $f(t,x)$, $\varphi(x)$, $b(t)$, $a(t)$; choose the initial approximation $y_0(t)$.

2. Solve direct problem (1) – (4).

3. Calculate the value of the functional $I(y)$; if $I(y) < \varepsilon$ then algorithm breaks and results are shown on the screen; if $I(y) > \varepsilon$ then move on to the point **4**.

4. Solve adjoint problem (6) – (9) and find $I'(y) = \psi(t,1)$.

5. Choose α as a constant.

6. Construct next approximation $y(t)$ by the formula:

$$y_{n+1}(t) = y_n(t) - \alpha_n\psi(t,1), \quad 0 < t < T. \tag{10}$$

Go to **2**.

Approximation of the direct and adjoint problems. [2], [4]

1. Direct problem (1) – (4) can be approximated as:

$$\frac{1}{h^2}u_{i-1}^{j+1} - \Big(\frac{2}{h^2} + \frac{1}{\tau}\Big)u_i^{j+1} + \frac{1}{h^2}u_{i+1}^{j+1} = -\Big(f_i^j + \frac{1}{\tau}u_i^j - \big(u_i^j\big)^3\Big),$$
$$j = 0, 1, \ldots, M-1, \quad i = 1, \ldots, N-1, \tag{11}$$

$$u_i^0 = \varphi_i, \quad i = 0, 1, \ldots, N, \tag{12}$$

$$u_0^j = u_1^j - b_j h, \quad j = M, M-1, \ldots, 1, 0, \tag{13}$$

$$u_N^j = u_{N-1}^j + y_j h, \quad j = M, M-1, \ldots, 1, 0. \tag{14}$$

2. Adjoint problem ((6) – (9) can be approximated as:

$$\frac{1}{h^2}\psi_{i-1}^{j} - \Big(\frac{2}{h^2} + \frac{1}{\tau}\Big)\psi_i^{j} + \frac{1}{h^2}\psi_{i+1}^{j} = -\frac{1}{\tau}\psi_i^{j+1},$$
$$j = M-1, M-2, \ldots, 1, 0, \quad i = 1, \ldots, N-1, \tag{15}$$

$$\psi_i^M = 0, \quad i = 0, 1, \ldots, N, \tag{16}$$

$$\psi_N^j = \psi_{N-1}^j + 0, \quad j = M, M-1, \ldots, 1, 0, \tag{17}$$

$$\psi_0^j = \psi_1^j + 2h\big(u_0^j - a_j\big), \quad j = M, M-1, \ldots, 1, 0. \tag{18}$$

3. Approximation of the functional. We approximate the value of the functional $I(y) = \int\limits_0^T \Big(u(t,0;y) - a(t)\Big)^2 dt$ by the conventional "rectangles formula":

$$I(y) \approx \left(\tau \sum_{j=0}^{M-1} \Big(u_0^j - a_j\Big)^2\right)^{1/2}. \tag{19}$$

Performing experiments

We use the developed numerical algorithm to solve the problem with the following set of functions:

$u(t,x) = e^{x-t}$ is a solution of (1) – (4),
$f(t,x) = e^{((p+1)(x-t))} - 2e^{(x-t)}$ is a free term of equation (1),
$\varphi(x) = e^x$ is initial temperature,
$b(t) = 2e^{-t}$ is a left boundary condition,
$a(t) = e^{-t}$ is an additional information,
$y(t) = e^{1-t}$ is an exact solution of the problem,
$y_0(t) = 2$ is an initial approximation.

Different degrees of nonlinearity are considered with the p-values of $\left\{\frac{1}{2},\ 2,\ 4\right\}$. We solve the problem for all chosen values of p with $T = 1$ and then with $T = 10$.

In all cases we take $M = 100$ and $N = 100$ (number of steps by t and x correspondingly) until otherwise is given. The parameters as ε, α as well as number of iterations and deviation of the solution from the exact values by norm of the space $\mathbf{H}$ are given in subscriptions to the figures.

Our goal is to analyze how different degrees of nonlinearity (the value of p is responsible for that) affects to numerical algorithm. Here we have two opposite factors that influence on the result.

First, we understand that the higher degrees of nonlinearity will affect negatively. Numerical algorithm will have bigger errors and that's why the obtained solution can be not so accurate.

Second, let consider the equation with odd degree of nonlinearity:

$$\partial_t u(t,x) = \partial_{xx} u(t,x) - u^{2m+1}(t,x) + f(t,x).$$

Let's multiply both sides of the equation by function $u(t,x)$ and after integration by the whole region $0 < t < T$, $0 < x < 1$ we get:

$$\int \partial_t u(t,x) u(t,x) = \int \partial_{xx} u(t,x) u(t,x) - \int u^{2m+2}(t,x) + \int f(t,x) u(t,x).$$

Using integration by parts and rearranging $\partial_t u(t,x) u(t,x)$:

$$\frac{1}{2} \int \frac{\partial}{\partial t} u^2(t,x) + \int \partial_x u^2(t,x) + \int u^{2m+2}(t,x) = \int f(t,x) u(t,x)$$

The left hand side of the equation contains the norm of the function u and that's why we expect good properties of the equation.

1. First Experiment. The following three figures 1, 2, 3 represent numerical solutions for different values of p and $T = 1$. You can see clearly how it affects on the solution. In all situations the solution $y(t) = e^{1-t}$ is the same, but the function $f(t,x) = e^{(p+1)(x-t)} - 2e^{x-t}$ (free term) is different.

As you can see, the degree of nonlinearity is responsible for the accuracy of the solution for the values of t which are close to zero. Weak convergence of the numerical algorithm near the point $t = 0$ can be explained easily. When direct problem (1) – (4) is solved numerically, error accumulates from $t = 0$ to $t = T$, so the worst solution we have for $t = T$. Then we solve the adjoint problem from $t = T$ to $t = 0$ and the gradient $I'(y) = \psi(t,1)$ has the worst values when $t = 0$. That's why we observe such discrepancy for the time values near $t = 0$.

A jump for $t = T$ was considered in [1], [2] and [3].

If the time horizon increases then it has negative impact on the accuracy of the solution. The following experiments illustrate this fact.

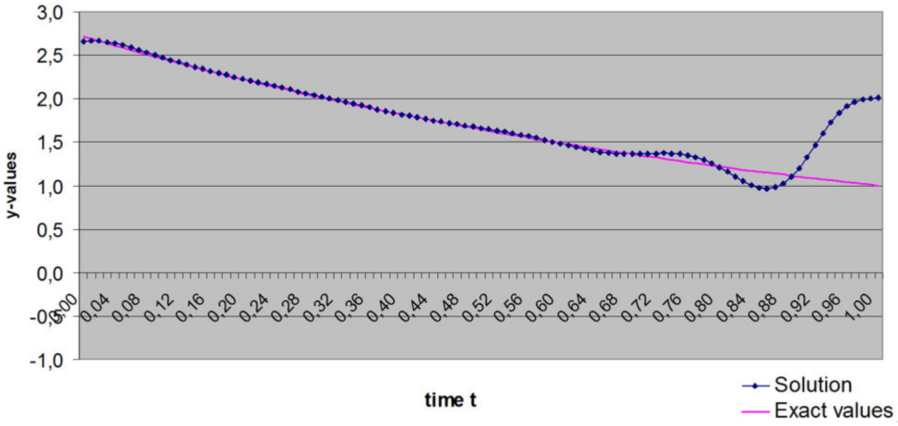

Figure 1 – Exact solution $y(t) = e^{1-t}$ and numerical solution for: $\varepsilon = 0.0003$, $\alpha = 10$, $p = 1/2$; $||y_n - y_{\text{exact}}||_{\mathbf{H}} = 0.230$.

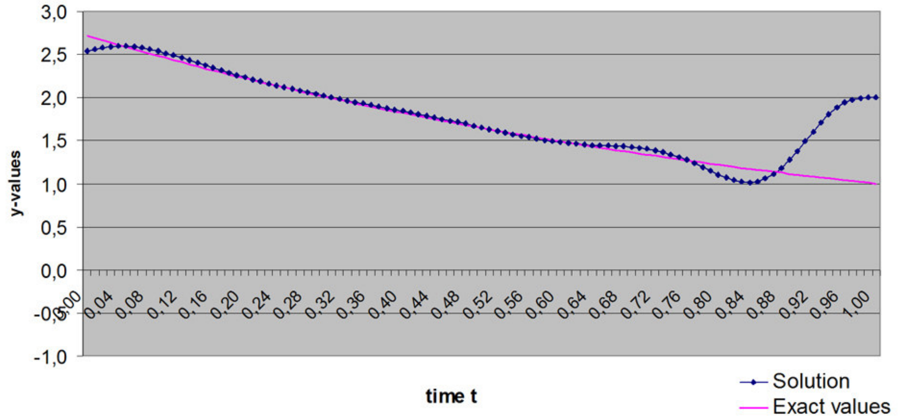

Figure 2 – Exact solution $y(t) = e^{1-t}$ and numerical solution for: $\varepsilon = 0.0005$, $\alpha = 50$, $p = 2$; $||y_n - y_{\text{exact}}||_{\mathbf{H}} = 0.247$.

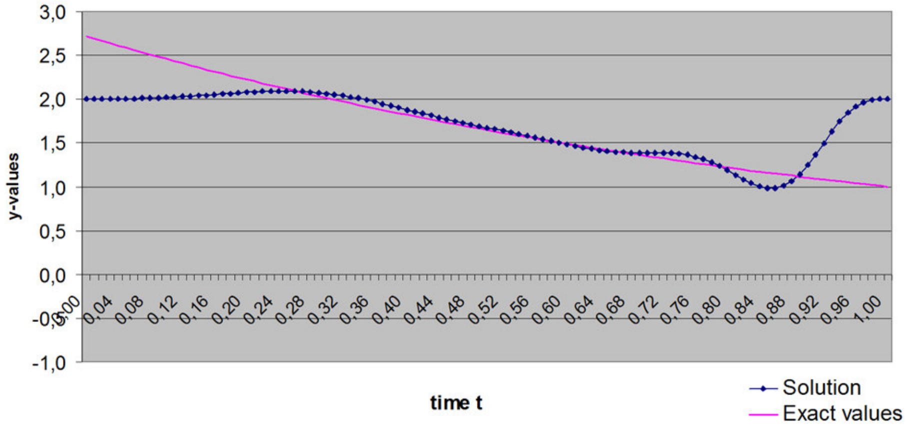

Figure 3 – Exact solution $y(t) = e^{1-t}$ and numerical solution for: $\varepsilon = 0.0004$, $\alpha = 1$, $p = 4$, $M = 200$; $||y_n - y_{\text{exact}}||_{\mathbf{H}} = 0.319$.

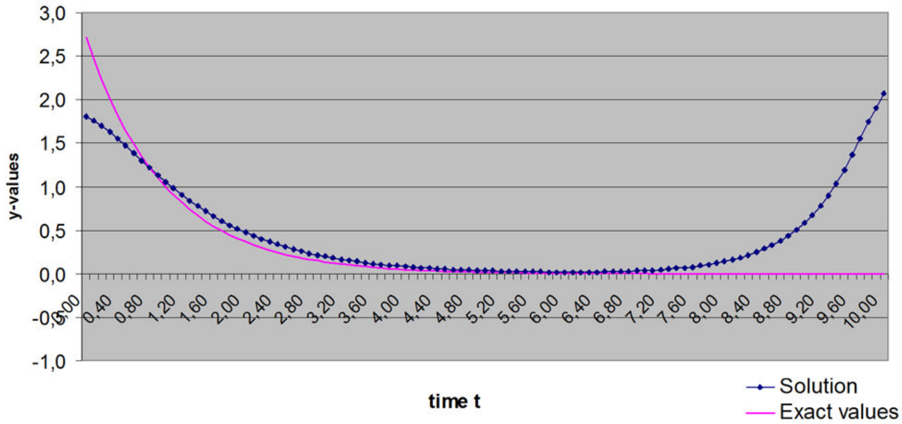

Figure 4 – Exact solution $y(t) = e^{1-t}$ and numerical solution for: $\varepsilon = 0.6$, $\alpha = 0.01$, $p = 1/2$; $||y_n - y_{\text{exact}}||_{\mathbf{H}} = 1.368$.

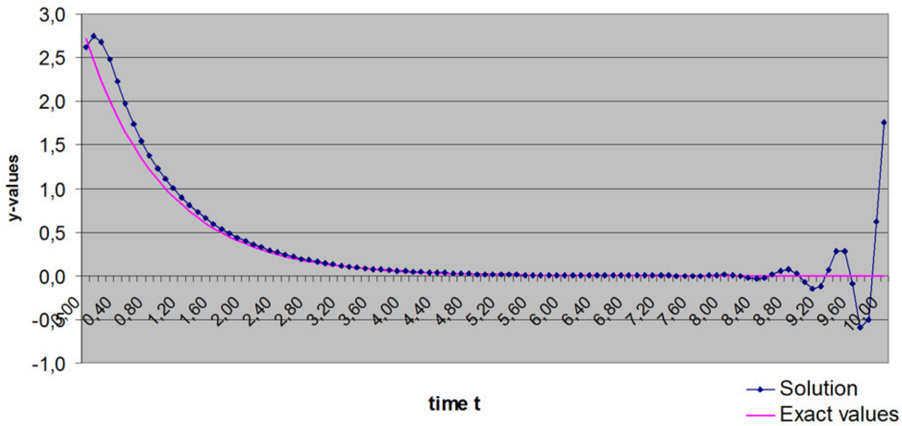

Figure 5 – Exact solution $y(t) = e^{1-t}$ and numerical solution for: $\varepsilon = 0.015$, $\alpha = 0.005$, $p = 2$; $||y_n - y_{\text{exact}}||_{\mathbf{H}} = 0.470$.

2. Second Experiment. For all next three figures 4, 5, 6, which represent a numerical solution in comparison with exact values, the value of $T = 10$.

As we expected, for the values close to $t = 0$ the numerical solution has very low precision. We do not pay attention to the jump in solution for the values close to $t = T$, the nature of this phenomena is described in [1], [2] and [3]. Also we emphasize on the fact that we used $M = 200$ for the last experiment. Because of if we take $M = 100$ (that is if we divide time horizon on 100 equal parts) then algorithm diverges.

It is a challenge to deal with parameters ε and α. Unavoidable error in numerical algorithm with a high degree of nonlinearity does not guarantee that the smaller the value of ε the more accurate solution you get. If the values of α are too small or too large the numerical algorithm can converge very slowly or diverge at all.

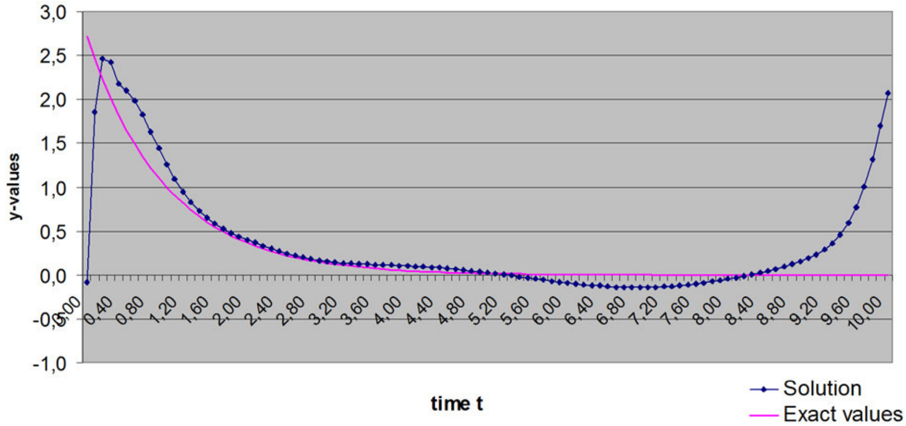

Figure 6 – Exact solution $y(t) = e^{1-t}$ and numerical solution for: $\varepsilon = 0.4$, $\alpha = 0.1$, $p = 4$, $M = 200$; $||y_n - y_{\text{exact}}||_{\mathbf{H}} = 1.783$.

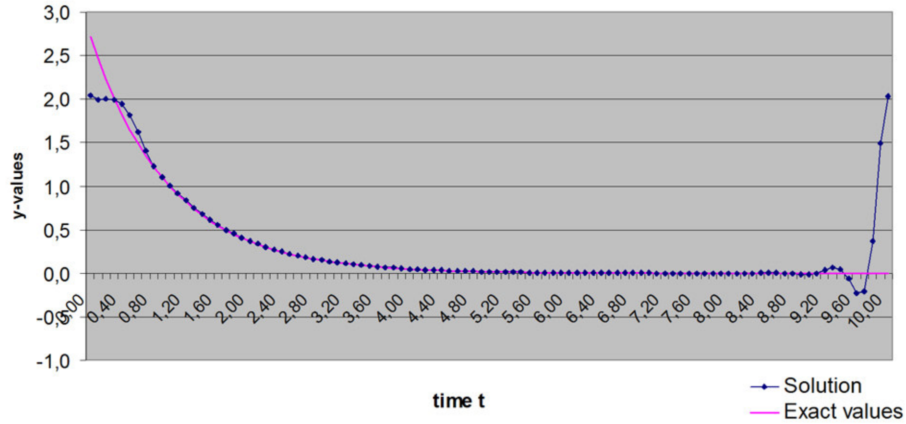

Figure 7 – Exact solution $y(t) = e^{1-t}$ and numerical solution for: $\varepsilon = 0.01$, $\alpha = 0.01$, $p = 4$, $M = 500$; $||y_n - y_{\text{exact}}||_{\mathbf{H}} = 0.686$.

3. Third Experiment.

The situation described in Figure 6 can be improved if the larger value of M is taken.

Figure 7 confirms all previous conclusions. The jump near the final values of time becomes very small in sense of $\mathbf{H}$ norm. Increasing of M (number of steps by variable t in numerical algorithm) accumulates less error near $t = 0$.

Conclusions

1. High degrees of nonlinearity of equation (1) causes bigger errors after applying numerical algorithms. To compensate that error we need to split time interval on larger number of steps. Accumulated errors cause the maximal discrepancy between numerical solution and exact values of function $y(t)$ for initial time moment, near $t = 0$. We give an explanation of this fact in First Experiment.

2. Choosing parameters ε and α is a challenge. If we take too small values of α, algorithm converges very slowly, on the other hand, if we take too large values, algorithm diverges. Because of unavoidable errors in algorithm the further decreasing of ε has no meaning and leads to increasing of number of iterations only.

Замечание 1 *For a recent time when using gradient methods in optimal control problems and inverse problems of mathematical physics a researcher can first find a discrete form of a system in order to make the algorithm more efficient. Then the researcher find a gradient of the functional and operate with it. [13]. These questions were also considered by author, [2], in the context of discussed problems but for the linear case. We could expect that in a problem with a exponential nonlinearity on a large time interval this method would be efficient.*

References

[1] *Becker J., Dagum L.* Particle simulation on heterogeneous distributed supercomputers // Concurrency-Practice and experience. – 1993. – №5(4). – C. 367–377.

[2] *Fougere D., Malyshkin V.* NumGrid middleware: MPI support for computational grids // Lecture Notes in Computer Science (PACT 2005). – 2005. – V.3606. – P. 313–320.

[3] *Diaz J., Munoz-Caro C., Nino A.A.* Survey of Parallel Programming Models and Tools in the Multi and Many-Core Era // IEEE Transactions on parallel and distributed systems. – 2012. – V.23(8). – P. 1369–1386.

[4] *Cappello F., Djilali S., Fedak G.* Computing on large-scale distributed systems: XtremWeb architecture, programming models, security, tests and convergence with grid // Future generation computer systems. – 2005. – V. 21(3). – P. 417–437.

[5] *Wang J., Liu Z.* Parallel Data Mining Optimal Algorithm of Virtual Cluster // International Conference on Fuzzy Systems and Knowledge. –2008. – V.5. – P. 358–362.

[6] *Pandey S., Buyya R.* Scheduling Workflow Applications Based on Multi-source Parallel Data Retrieval in Distributed Computing Networks // Computer journal. – 2012. – V. 55(11). – P. 1288–1308.

[7] *Liu H., Orban D.* GridBatch: Cloud Computing for Large-Scale Data-Intensive Batch Applications // CCGRID. – 2008. – V. 1. – P.295–305.

[8] *Valilai O., Houshmand M.* A collaborative and integrated platform to support distributed manufacturing system using a service-oriented approach based on cloud computing paradigm // Robotics and computer-integrated manufacturing. –2013. –V.29(1). – P.110–127.

[9] *Ahmed-Zaki D., Dobrowolski G., Kumalakov B.* Peer-to-Peer MapReduce Platform // Proceedings of the 5th International Conference on Agents and Arti?cial Intelligence. (ICAART 2013). – 2013. – V. 2. – P. 565–570.

[10] *Akhmed-Zaki D., Kumalakov B.* Composite Peer-to-Peer MapReduce System // Proceedings of the International Conference on New Trends in Information and Communication Technologies (ICTT 2013). – 2013. – P.44—49.

[11] *Akhmed-Zaki D.Zh., Kumalakov B.A.* Solving complex iterative tasks using intellectual load distribution and MPI // Вестник Национальной инженерной академии Республики Казахстан. – 2014. – №2(52). –P. 25–31.