Denis Constales; Jozef Kačur

On the solution of inverse problems for generalized oxygen consumption

Applications of Mathematics, Vol. 46 (2001), No. 2, 145--155

Persistent URL: http://dml.cz/dmlcz/134461

ON THE SOLUTION OF INVERSE PROBLEMS FOR GENERALIZED OXYGEN CONSUMPTION

D. Constales, Gent, J. Kačur, Bratislava

(Received July 9, 1999)

Abstract. We present the solution of some inverse problems for one-dimensional free boundary problems of oxygen consumption type, with a semilinear convection-diffusion-reaction parabolic equation. Using a fixed domain transformation (Landau's transformation) the direct problem is reduced to a system of ODEs. To minimize the objective functionals in the inverse problems, we approximate the data by a finite number of parameters with respect to which automatic differentiation is applied.

Keywords: oxygen consumption, inverse problems, automatic differentiation

MSC 2000: 35R35, 49M15, 49N50

1. Introduction

Let us consider the direct parabolic problem

$$u_t = \partial_x\bigl(a(u)u_x + b(x)u\bigr) - F(u) \tag{1}$$

in $x \in \bigl(0, s(t)\bigr)$, $t > 0$, $s(t) > 0$, with the interface conditions

$$u\bigl(s(t), t\bigr) = 0, \qquad u_x\bigl(s(t), t\bigr) = 0, \tag{2}$$

the boundary condition

$$-\bigl(a(u)u_x + b(x)u\bigr)_{x=0} = q(t) \tag{3}$$

and the initial conditions

$$u(x, 0) = u_0(x), \qquad s(0) = s_0, \tag{4}$$

where $u_0(x) \geqslant 0$ and $s_0 > 0$.

If $b=0$ and $F(u)=m$ (a constant), this is the well-known oxygen consumption problem with nonlinear diffusion. The second condition in Eq. (2) could equivalently be written as

$$\big(a(u)u_x+b(x)u\big)\big|_{x=s(t)}=0,$$

which implies that the flux is zero on the boundary $x=s(t)$. The interface is implicitly given in (2) and this makes the problem difficult from the numerical point of view.

We assume that $a(u),F(u)\geqslant q>0$ are continuous and that $a,b\in C^1$. The existence and uniqueness of the solution of (1)–(4) can be obtained in terms of the variational solution of the corresponding variational inequality,

$$(u_t,v-u)+\big(a(u)u_x+b(x)u,v_x-u_x\big)+\big(q(t),v-u\big)_{\Gamma_1}\geqslant-(F(u),v-u)\ \ \forall v\in L_2(I,K),$$

where $I=(0,T)$ $(T>0)$ is the time interval,

$$K=\{v\in W_2^1(\Omega)\colon\ v\geqslant 0\text{ in }\Omega\},$$
$$(u,v)=\int_\Omega uv\,\mathrm{d}x,\quad \Omega=(0,l),\quad (u,v)_{\Gamma_1}=u(0)v(0),$$

and $s_0<l$, l being sufficiently large. For details see [1], [2].

We will be concerned here with the numerical approximation of (1)–(4) and inverse problems (1)–(5). We assume that the solution u is smooth in the interval $[0,s(t)]$ and positive in $(0,s(t))$. This gives rise to structural assumptions concerning $F(u)$, $b(x,t)$ and $a(u)$, as special choices of this data can cause the solution u to vanish inside the interval $(0,s(t))$. In such cases our numerical results will not correspond to (1)–(4). An example is the choice $a(u)=1$, $b(x)=x-\frac12$, $F(u)=1+\sqrt{u}$ and $u_0=\frac12(1-x)^2$.

For the inverse problems, we will try to reconstruct the data $a(u)$, $F(u)$ from measurements on the boundary $x=0$ or from measurements of $s(t)$, while $q(t)$ will be reconstructed from the profile $u(x,T)$ at a specified time T.

The objective functionals

$$G_1(p)=\int_0^T\big(u_p(0,t)-\varphi_1(t)\big)^2\,\mathrm{d}t+\alpha J(p),\tag{5}$$

$$G_2(p)=\int_0^T\big(s_p(t)-\varphi_2(t)\big)^2\,\mathrm{d}t+\alpha J(p)\tag{6}$$

and

$$G_3(p)=\big(s_p(T)-\hat{s}\big)^2+\lambda\int_0^{s(T)}\big(u_p(x,s(T))-\varphi_3(x)\big)^2\,\mathrm{d}x+\alpha\int_0^T|q_p(t)|\,\mathrm{d}t\tag{7}$$

will be used in the inverse problems, to be minimized w.r.t. $p = (p_1, \dots, p_m)$, the parameters of $a(u)$, $b(x)$, $F(u)$ or $q(t)$. Here $\varphi_1(t)$ is obtained from the measurements at $x = 0$, $\varphi_2(t)$ from the measurements of $s(t)$, $\varphi_3(x)$ is the target profile $u(x, T)$ and $\hat{s}$ its boundary position, $u_p(x, t)$ is the solution of the corresponding direct problem, $s_p(t)$ the corresponding interface position, and $J(p)$ is a convex function in p, which is multiplied by a regularization parameter α. (This is the Tichonof-type regularization of this ill-posed inverse problem.) The parameter α is related to the error in the measurements at $x = 0$.

In Section 2 we present an approximate solution of the direct problem, which is numerically effective for the purposes of the inverse problem.

In the minimization problems (5)–(7) we use a Newton-Raphson minimization procedure where the corresponding first and second derivatives of G with respect to p are obtained by automatic differentiation (cf. e.g. [3]) and implemented in the well-known ODE solver LSODA (cf. [4], [5]), using our C++ language port and adaptation to automatic differentiation, LSODA-C. The point of automatic differentiation is that it allows the ODE solver to generate transparently the derivatives w.r.t. the parameters, thus offering a very versatile way of solving inverse problems, while obviating the need of constructing and implementing the full adjoint system.

2. Solution of the direct problem

Movement of the boundary.

We first use the transformation $w = \sqrt{u}$ in (1), which after straightforward simplification gives

$$w_t = a(w^2)w_{xx} + b(x)w_x + \left(\frac{1}{2}b_x(x) + 2a'(w^2)w_x^2\right)w + \frac{1}{2w}[2a(w^2)w_x^2 - F(w^2)] \tag{8}$$

along with the interface condition

$$w\big(s(t), t\big) = 0. \tag{9}$$

Assuming that our solution w is smooth up to the free boundary $x = s(t)$, the numerator of the rightmost term (in brackets) on the RHS of Eq. (8) must equal zero at $x = s(t)$ because of (9). Thus we obtain the second interface condition in w:

$$w_x\big(s(t), t\big) = -\sqrt{\frac{F(0)}{2a(0)}}, \tag{10}$$

the minus sign following from the fact that $w \geqslant 0$.

Differentiating (9) totally w.r.t. t yields the formula

$$s'(t) = -\lim_{x\to s(t)^-} \frac{w_t(x,t)}{w_x(x,t)}. \tag{11}$$

Now we substitute (10) for the denominator in this expression, while for $w_t(x,t)$ we take the RHS of (8). Since $w\big(s(t),t\big) = 0$ we can use l'Hospital's rule when considering the rightmost term in (8):

$$\begin{aligned}
&\lim_{x\to s(t)^-} \frac{1}{2w}[2a(w^2)w_x^2 - F(w^2)] \\
&\quad = \lim_{x\to s(t)^-} \frac{4a'(w^2)ww_x^3 + 4a(w^2)w_x w_{xx} - 2F'(w^2)ww_x}{2w_x} \\
&\quad = 2a(0)w_{xx}\big(s(t),t\big).
\end{aligned}$$

Combining this with the other terms on the RHS of (8) we obtain from (11) the formula for the movement of the boundary:

$$s'(t) = \sqrt{\frac{18a^3(0)}{F(0)}} w_{xx}\big(s(t),t\big) - b\big(s(t)\big). \tag{12}$$

Landau's transformation.

Now we use the transformation $y = x/s(t)$, writing $\overline{w}(y,t) = w(x,t)$; this gives for (12) that

$$s'(t) = \sqrt{\frac{18a^3(0)}{F(0)}}\,\frac{\overline{w}_{yy}(1,t)}{s^2(t)} - b\big(s(t)\big), \tag{13}$$

and for (8),

$$\begin{aligned}
\overline{w}_t = {}& \frac{a(\overline{w}^2)}{s^2(t)}\overline{w}_{yy} + \frac{b\big(ys(t)\big) + ys'(t)}{s(t)}\,\overline{w}_y + \left(\frac{1}{2}b'\big(ys(t)\big) + \frac{2a'(\overline{w}^2)\overline{w}_y^2}{s^2(t)}\right)\overline{w} \\
&+ \frac{1}{2\overline{w}}\left[2\frac{a(\overline{w}^2)}{s^2(t)}\overline{w}_y^2 - F(\overline{w}^2)\right].
\end{aligned} \tag{14}$$

In terms of $\overline{w}$ the boundary condition (3) is given by

$$-\Big(a(\overline{w}^2(0,t))\frac{{\overline{w}_y}^2(0,t)}{s(t)} + b(0)\overline{w}^2(0,t)\Big) = q(t)$$

or, equivalently,

$$\overline{w}_y(0,t) = -s(t)\,\frac{q(t) + b(0)\overline{w}^2(0,t)}{2a\big(\overline{w}^2(0,t)\big)\overline{w}(0,t)}. \tag{15}$$

Finally, the interface conditions (9) and (10) correspond to

$$\overline{w}(1,t)=0, \qquad \overline{w}_y(1,t)=-s(t)\sqrt{\frac{F(0)}{2a(0)}}, \tag{16}$$

and the initial conditions (16) to

$$\overline{w}(y,0)=\sqrt{u_0(ys_0)}, \qquad s(0)=s_0. \tag{17}$$

The problem is thereby reduced to the solution of (13–17) over $y\in(0,1)$ for $t>0$ and for as long as $s(t)>0$.

Space discretization.

Splitting the unit interval $y\in(0,1)$ into N subintervals (not necessarily of equal length), we introduce nodal points at positions $0=y_0<y_1<\ldots<y_N=1$ and the corresponding values for the discretized $\overline{w}(y_k,t)$ denoted by $C_0(t),\ldots,C_N(t)$.

We use second-order interpolation formulas to approximate the differential operators occurring in the system of PDEs, and to express the boundary conditions in the discretizations.

Recall that if x_1, x_2, x_3 are three different values, the Lagrange interpolating polynomial p of degree at most two, uniquely defined by $p(x_i)=f_i$, $i=1,2,3$, is given explicitly by

$$p(x)=\frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)}f_1+\frac{(x-x_3)(x-x_1)}{(x_2-x_3)(x_2-x_1)}f_2+\frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}f_3.$$

Its second derivative will be denoted by $\mathcal{D}_2$ and is given by

$$\begin{aligned}\mathcal{D}_2(x_1,f_1,x_2,f_2,x_3,f_3)=&\frac{2}{(x_1-x_2)(x_1-x_3)}f_1+\frac{2}{(x_2-x_3)(x_2-x_1)}f_2\\&+\frac{2}{(x_3-x_1)(x_3-x_2)}f_3.\end{aligned}$$

The first derivative $p'(x_2)$ will be denoted by $\mathcal{D}_1$ and is given by

$$\begin{aligned}\mathcal{D}_1(x_1,f_1,x_2,f_2,x_3,f_3)=&\frac{x_2-x_3}{(x_1-x_2)(x_1-x_3)}f_1+\frac{2x_2-x_1-x_3}{(x_2-x_3)(x_2-x_1)}f_2\\&+\frac{x_2-x_1}{(x_3-x_1)(x_3-x_2)}f_3.\end{aligned}$$

The interface conditions (16) are expressed by setting $C_N(t)=0$ identically, and by writing $C_{N-1}(t)$ in terms of $C_{N-2}(t)$ through the solution of the linear equation

$$\mathcal{D}_1\big(y_{N-2},C_{N-2}(t),1,0,y_{N-1},C_{N-1}(t)\big)=-s(t)\sqrt{\frac{F(0)}{2a(0)}}.$$

Consequently there are $N-1$ independent discretization values $C_0(t), \ldots, C_{N-2}(t)$ to be simulated, along with $s(t)$. The initial conditions (17) are used to initialize them:

$$C_k(0) = \sqrt{u_0(y_k s_0)}, \quad k = 0, \ldots, N-2, \qquad s(0) = s_0.$$

For the boundary condition (15) a fictive point $y_{-1} = -y_1$ is introduced, with the corresponding fictive $\overline{w}$ value $C_{-1}(t)$ given by

$$C_{-1}(t) = C_0(t) + y_1 s(t)\, \frac{q(t) + b(0)\overline{w}^2(0,t)}{a(\overline{w}^2(0,t))\overline{w}(0,t)}.$$

The discretization of (14) at y_k, $k = 0, \ldots, N-2$, is achieved by substituting $C_k(t)$ for $\overline{w}$, $C_k'(t)$ for $\overline{w}_t$, $\mathcal{D}_1\big(y_{k-1}, C_{k-1}(t), y_k, C_k(t), y_{k+1}, C_{k+1}(t)\big)$ for $\overline{w}_y$ and $\mathcal{D}_2\big(y_{k-1}, C_{k-1}(t), y_k, C_k(t), y_{k+1}, C_{k+1}(t)\big)$ for $\overline{w}_{yy}$.

Finally, the four-point interpolation formula for the second derivative at the first point,

$$\begin{aligned}\mathcal{D}_{4,2}(x_1, f_1, x_2, f_2, x_3, f_3, x_4, f_4) = {} & 2\frac{3x_1 - x_4 - x_3 - x_2}{(x_1 - x_2)(x_1 - x_3)(x_1 - x_4)} f_1 \\ & - 2\frac{2x_1 - x_4 - x_3}{(x_1 - x_2)(x_2 - x_3)(x_2 - x_4)} f_2 \\ & + 2\frac{2x_1 - x_4 - x_2}{(x_1 - x_3)(x_2 - x_3)(x_3 - x_4)} f_3 \\ & - 2\frac{2x_1 - x_3 - x_2}{(x_1 - x_4)(x_2 - x_4)(x_3 - x_4)} f_4,\end{aligned}$$

is used to discretize (13) by substituting

$$\mathcal{D}_{4,2}\big(1, 0, y_{N-3}, C_{N-3}(t), y_{N-2}, C_{N-2}(t), y_{N-1}, C_{N-1}(t)\big)$$

for $\overline{w}_{yy}$. (Using a three-point formula there will not guarantee the accurate convergence of the simulations.)

The resulting system of ODEs is integrated using LSODA in the form

$$\begin{gathered}\dot{C}(t) = f\big(t, C(t), a, b, q, F\big) \\ \dot{s}(t) = g\big(t, C(t), a, b, F\big) \\ C(0) = C_0, \quad s(0) = s_0, \quad C \text{ denoting } (C_0, \ldots, C_{N-2}).\end{gathered}$$

3.1. Experiment 1.

In Fig. 1 we present the time evolution of $s(t)$ and the evolution of $u(x,t)$ for the

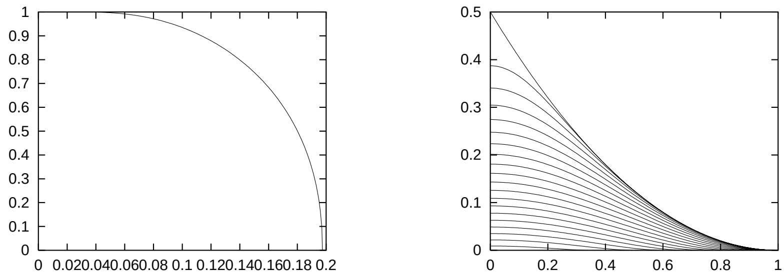

Figure 1. Simulation results for the classical oxygen diffusion problem. Left: $s(t)$ vs. t. Right: $u(x,t)$ vs. x for $t = 0, 0.01, \ldots, 0.19$.

classical oxygen-diffusion problem, where $a(u) = 1$, $F(u) = 1$, $q(t) = 0$, $b(x) = 0$, $s(0) = 1$ and $u_0(x) = \frac{1}{2}(1-x)^2$. The effectiveness of our method can be judged from the steady improvements in the estimated extinction time T (i.e., the smallest time at which $s(T) = 0$, all oxygen having been consumed). In the table below we present the numerical estimate of extinction time $T(N)$ using N grid points. The third column represents the Richardson approximation of $T(N)$, assuming an error term of order α/N^2.

N	$T(N)$	$(4T(N) - T(N/2))/3$
5	0.199 332 162	—
10	0.197 876 096	0.197 390 741
20	0.197 544 467	0.197 433 924
40	0.197 462 316	0.197 434 932
80	0.197 441 816	0.197 434 983
160	0.197 436 694	0.197 434 987

3.2. Experiment 2. In Fig. 2 we show the time evolution of the interface for a peridically changing flux $q(t)$, following a block wave with frequency ν ($\{\cdot\}$ denoting the fractional part of a number):

$$q(t) = \begin{cases} Q & \text{if } 0 \leqslant \{\nu t\} < \frac{1}{2}, \\ 0 & \text{otherwise.} \end{cases}$$

The other functions and parameters are as in Experiment 1. Note the period tripling of this nonlinear system.

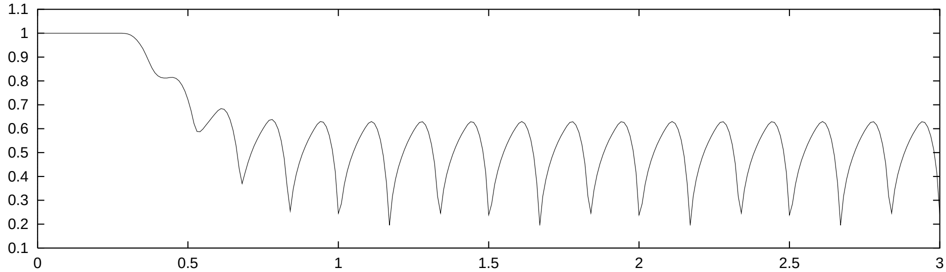

Figure 2. Simulation results for a periodically changing flux: $s(t)$ vs. t. Parameter choices: $Q = 1$, $\nu = 1/6$.

3.3. Experiment 3.

In Fig. 3 we present the time evolution of $s(t)$ and the evolution of $u(x,t)$ for a problem where $a(u) = 1$, $F(u) = 1$, $q(t) = 0$, $s(0) = 1$ and $u_0(x) = \frac{1}{2}(1-x)^2$ as before, but $b(x) = 1$, so that a convective term appears, which is visible in the rightward movement of the maximum. The extinction time is noticeably smaller (0.152 269 678), because the convection slows down the movement of the boundary, which means that more oxygen is consumed per unit of time.

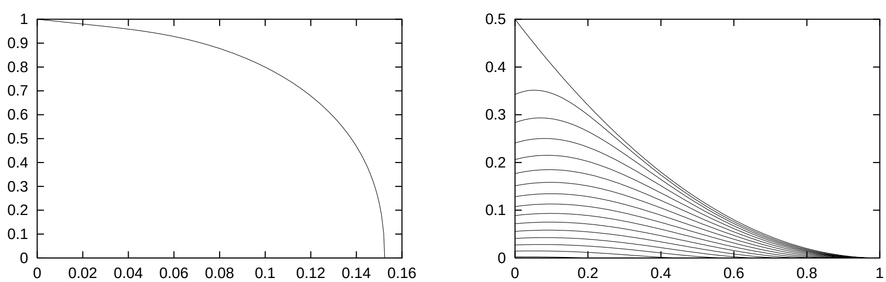

Figure 3. Simulation results for the classical oxygen diffusion problem with a convection term $b(x) = 1$. Left: $s(t)$ vs. t. Right: $u(x,t)$ vs. x for $t = 0, 0.01, \ldots, 0.15$.

4. Numerical results for the inverse problem

To verify stability of the solution of the ill-posed inverse problem, we add a noise term $\delta\eta$ to the measurements $u(0,t)$ and $s(t)$, where $\delta > 0$ is the amplitude and η is a normally distributed random variable.

4.1. Experiment 4.

• Determination of $F(u) = 1 + 2u$ from $u(0,t)$ data:

δ	α	$F(0)$	$F(0.125)$	$F(0.25)$	$F(0.375)$	$F(0.5)$
0.001	0	1.00317	1.24480	1.50646	1.75815	1.97864
0.01	0	0.99829	1.21236	1.55421	1.84780	1.74569
0.01	0.01	0.96433	1.23728	1.53080	1.82706	2.12228
0.01	1	0.95516	1.24267	1.53048	1.81837	2.10625

• Determination of $F(u) = 1 + 2u$ from $s(t)$ data:

δ	α	$F(0)$	$F(0.125)$	$F(0.25)$	$F(0.375)$	$F(0.5)$
0.00001	0	1.00000	1.25003	1.50027	1.74660	2.02083
0.0001	0	1.00010	1.25031	1.50313	1.71217	2.24192
0.001	0	1.00615	1.24218	1.59714	2.10379	−5.23543
0.01	0	0.61521	2.61734	3.41689	5.89838	−19.48335
0.01	1	1.00766	1.24720	1.48626	1.72519	1.96412
0.01	0.01	1.00605	1.25849	1.47058	1.67102	1.87060

4.2. Experiment 5.

• Determination of $a(u) = 1 + 2u$ from $u(0,t)$ data:

δ	α	$a(0)$	$a(0.125)$	$a(0.25)$	$a(0.375)$	$a(0.5)$
0.001	0	0.99851	1.24643	1.50873	1.73599	2.03298
0.01	0	0.98899	1.21313	1.59110	1.59447	2.39829
0.01	0.01	0.95638	1.25416	1.53965	1.79510	2.04959
0.01	1	0.97050	1.25174	1.53276	1.81336	2.09393

• Determination of $a(u) = 1 + 2u$ from $s(t)$ data:

δ	α	$a(0)$	$a(0.125)$	$a(0.25)$	$a(0.375)$	$a(0.5)$
0.001	0	0.90231	1.33501	1.43210	2.19421	4.76297
0.01	0	1.18087	0.80309	2.51533	4.59328	53.45683
0.01	0.01	1.04412	1.22556	1.43006	1.62805	1.84591
0.01	1	1.04669	1.22984	1.41337	1.59684	1.78053

4.3. Experiment 6.

• Determination of $(t) = 1 + 2t$ from $u(0, t)$ data, $0 \leqslant t \leqslant 0.5$:

δ	α	$q(0)$	$q(0.125)$	$q(0.25)$	$q(0.375)$	$q(0.5)$
0.01	0	0.99394	1.24985	1.50164	1.74852	2.00558

• Determination of $q(t) = 1 + 2t$ from $s(t)$ data, $0 \leqslant t \leqslant 0.5$:

δ	α	$q(0)$	$q(0.125)$	$q(0.25)$	$q(0.375)$	$q(0.5)$
0.00001	0	0.99999	1.24999	1.50001	1.74991	2.00156
0.0001	0	0.99994	1.24996	1.50019	1.74919	2.01569
0.001	0	0.99950	1.24964	1.50195	1.74199	2.15694
0.01	0	0.99501	1.24646	1.51953	1.67023	3.56892
0.01	0.01	0.99065	1.25359	1.49897	1.75828	2.03288
0.01	1	0.99527	1.24927	1.50265	1.75605	2.00960

4.4. Experiment 7.

• Determination of $b(x) = 0.5 - x$ from $u(0, t)$ data:

δ	α	$b(0)$	$b(0.25)$	$b(0.5)$	$b(0.75)$	$b(1)$
0.00001	0.01	0.50006	0.25016	−0.00044	−0.24751	−0.62031
0.01	0.01	0.47164	0.12795	−0.17951	−0.47353	−0.76762
0.01	1	0.47571	0.15281	−0.16967	−0.49202	−0.81436

• Determination of $b(x) = 0.5 - x$ from $s(t)$ data:

δ	α	$b(0)$	$b(0.25)$	$b(0.5)$	$b(0.75)$	$b(1)$
0.001	0	0.49950	0.24796	0.00388	−0.26165	−0.42655
0.01	0.01	0.50016	0.25101	0.02188	−0.19705	−0.40906
0.01	1	0.49782	0.25887	0.02049	−0.21751	−0.45541

4.5. Experiment 8.

Finally, we consider a problem in which the reconstruction is not based on data at different times, but on the profile $u(x, T)$ at some specified time T: given a target profile $\hat{u}$ with interface position $\hat{s}$, what inward flux $q(t)$ should be applied during $t \in (0, T)$ in order that the solution $u(x, T)$ approximate $\hat{u}(x)$?

In the following experiment, $a = 1$, $b = 0$, $F = 1$, $T = 0.1$ and we are trying to recover $q(t) = q_0 + q_1 t = 15 - 200t$ from the perturbed data. As before, δ is the amplitude of a normally distributed perturbation of the target profile data $\hat{u}(x)$ at the sampling points.

For $\delta = 0.03$ we recover $q = 15.2 - 204t$ without regularization, from the points depicted in Fig. 4. (Note that this solution and the original one are almost indistinguishable.)

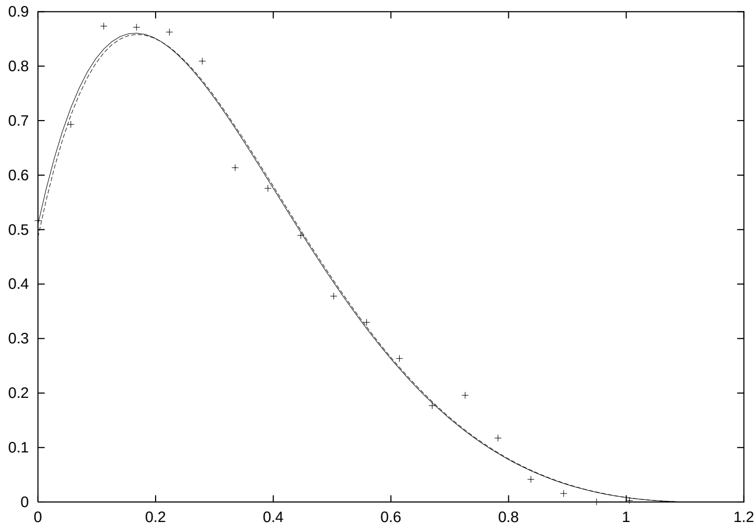

Figure 4. Reconstruction of $q(t)$ from $u(x,T)$: $u(x,T)$ and perturbed data points vs. x.

References

[1] *H. W. Alt, S. Luckhaus*: Quasilinear elliptic-parabolic differential equations. Math. Z. *183* (1983), 311–341.

[2] *J. Crank*: Free and Moving Boundary Problems. Oxford Science Publications, Clarendon Press, Oxford, 1984.

[3] *A. Griewank, G. F. Corliss*: Automatic Differentiation of Algorithms: Theory, Implementation, and Application. SIAM, Philadelphia, 1991.

[4] *A. C. Hindmarsh*: `ODEPACK`, a systematized collection of ODE solvers. In: Scientific Computing (R. S. Stapleman *et al.* (eds)). North-Holland, Amsterdam, 1983, pp. 55–64.

[5] *L. R. Petzold*: Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. SIAM J. Sci. Comput. *4* (1983), 136–148.

Authors' addresses: *D. Constales*, Dept. Pure Math. & Comp. Alg., Gent Univ., Galglaan 2, 9000 Gent, Belgium, e-mail: `denis.constales@rug.ac.be`; *J. Kačur*, Dept. Num. Math. and Opt., Comenius Univ., 842 15 Bratislava, Slovakia, e-mail: `kacur@fmph.uniba.sk`.