The integration of decomposition / factorization method in abstract operator equations in CAS software *Mathematica*

K.D. Tsilika

Abstract. This paper suggests the integration of a new, universal decomposition method for the class of abstract operator equations $B_1x = f$ in CAS software *Mathematica*, in order to have the exact solutions in an arbitrary Banach space. The application of the analytical solution method in numerical examples is a tedious, complicated and time-consuming task. Combining the algorithmic structure of the theoretical solution method and the symbolic capabilities of an established CAS software, the formation of the exact solutions of ordinary or partial integro-differential equations with nonlocal and initial boundary conditions becomes an easy task.

Introduction

We investigate the problem $B_1x = f$ where operator B_1 can be written as a product of two other correct operators B_0, B i.e. $B_1 = B_0B$. The methodology for the formation of the unique solution is developed based on previous work of [1, 2, 3, 4, 5].

Theorem. Let X and Z be Banach spaces, $Z \subseteq X$, the vectors $G = (g_1, ..., g_m)$, $G_0 = (g_1^{(0)}, ..., g_m^{(0)})$, $S_0 = (s_1^{(0)}, ..., s_m^{(0)}) \in X^m$, the components of the vectors $F = col(F_1, ..., F_m)$ and $\Phi = col(\Phi_1, ..., \Phi_m)$ belong to X^* and Z^*, respectively, and the operators $B_0, B, B_1 : X \to X$ be defined by

$$B_0x = A_0x - G_0\Phi(x) = f, \qquad D(B_0) = D(A_0) \subset Z, \tag{1}$$

$$Bx = Ax - GF(Ax) = f, \qquad D(B) = D(A), \tag{2}$$

$$B_1x = A_0Ax - S_0F(Ax) - G_0\Phi(Ax) = f, \quad D(B_1) = D(A_0A), \tag{3}$$

where A_0 and A are linear correct operators on X and $G \in D(A_0)^m$. Then the following statements are satisfied:

(i) If

$$S_0 \in R(B_0)^m \quad \text{and} \quad S_0 = B_0G = A_0G - G_0\Phi(G), \tag{4}$$

then the operator B_1 can be decomposed in $B_1 = B_0B$.
(ii) If in addition the components of the vector $F = col(F_1, ..., F_m)$ are linearly independent elements of X^* and since the operator B_1 can be decomposed in $B_1 = B_0B$, then (4) is fulfilled.
(iii) If the operator B_1 can be decomposed in $B_1 = B_0B$ then B_1 is correct if and only if the operators B_0 and B are correct which means that

$$\det L_0 = \det[I_m - \Phi(A_0^{-1}G_0)] \neq 0 \quad \text{and} \quad \det L = \det[I_m - F(G)] \neq 0. \tag{5}$$

(iv) If the operator B_1 has the decomposition in $B_1 = B_0B$ and is correct, then the unique solution of (3) is

$$\begin{aligned} x = B_1^{-1}f &= A^{-1}A_0^{-1}f + A^{-1}GL^{-1}F(A_0^{-1}f) \\ &+ \left[A^{-1}A_0^{-1}G_0 + A^{-1}GL^{-1}F(A_0^{-1}G_0)\right] L_0^{-1}\Phi(A_0^{-1}f). \end{aligned} \tag{6}$$

1. The Solution Algorithm

In *Mathematica*' s environment, we program the solution method in 3 steps.

Step 1. Obtain a description of the problem. Define the functions and the operators $f, F, \Phi, G_o, S_o, G, A^{-1}, A_o^{-1}$ involved in the abstract operator equation $B_1x = f$, with B_1 defined as in (3).

Step 2. Check the correctness criteria for B_0, and B as given in (5).

Step 3. If the operator B_1 has the decomposition $B_1 = B_0B$ and is correct, then the unique solution of (3) is (6).

2. Working with Operators in *Mathematica*

Think of an expression like f[x] as being formed by applying an operator f to the expression x. An expression like f[g[x]] is the result of composing the operators f and g, and applying the result to x. In *Mathematica*, we set f to be a pure function by the following formula ([6]):
f = (3 + #)&
Input: f[a], f[b]
Output: 3 + a, 3 + b

3. Performance and Code Efficiency

The core part of the code for the automated solving of (3) is given in the figures below. For the abstract operator equations of examples 1,2 we present the input needed to define the problem, the input to check the correctness of B_0, and B, and the input that generates the unique solution (see figures 1, 2 correspondingly).

Define the operators of the initial problem

```
inverseA = ∫_0^t (t - s) * # ds + 1 / 4 * ∫_0^π (2 * s - 3 * t - π / 2) * # ds &
inverseAo = ∫_0^t # ds - 1 / 2 * ∫_0^π # ds &
F = ∫_0^π t^2 * # dt &; Φ = ∫_0^(π/2) (t + 1) * # dt &;
f = Sin[2 * s]; So = Sin[s]; Go = Cos[s];
```

Check correctness criterion for Bo

```
Lo = Det[IdentityMatrix[1] - Φ[inverseAo[Go]]]
```

Compute operator G

```
G = inverseAo[So] + inverseAo[Go] * (-1) * Φ[inverseAo[So]]
```

Check correctness criterion for B

```
L = Det[IdentityMatrix[1] - F[G]]
```

Compute the exact solution

```
inverseAinverseAo = inverseA[inverseAo[#] /. t → s] &
inverseAinverseAo[f] + inverseA[G /. t → s] * L^(-1) * F[inverseAo[f]] +
  (inverseAinverseAo[Go] + inverseA[G /. t → s] * L^(-1) * F[inverseAo[Go]]) *
   Lo^(-1) * Φ[inverseAo[f]]
```

FIGURE 1. *Mathematica* codes for solving Example 1

Example 1. Let $B_1 : C[0,\pi] \to C[0,\pi]$ the operator corresponding to the problem:

$$x'''(t) - \sin t \int_0^{\pi} t^2 x''(t)dt - \cos t \int_0^{\pi/2} (t+1)x''(t)dt = \sin 2t, \tag{7}$$
$$x(0) + x(\pi) = 0, \quad x'(0) + 3x'(\pi) = 0, \quad x''(0) + x''(\pi) = 0,$$

Example 2. Let $\Pi = \{(t,s) \in \mathbb{R}^2 : 0 \le t, s \le 1\}$. An operator $B_1 : C(\Pi) \to C(\Pi)$ corresponds to the problem:

$$x''_{ts}(t,s) - t^3 s \int_0^1 \int_0^1 s^2 x'_t(t,s)dtds - ts^2 \int_0^1 \int_0^1 t x'_t(t,s)dtds = 5t^2 + s, \tag{8}$$
$$x'_t, x''_{ts} \in C(\Pi), \quad x(0,s) = s^2 \int_0^1 \int_0^1 x(t,s)dtds,$$
$$x'_t(t,0) = t \int_0^1 \int_0^1 s x'_t(t,s)dtds,$$

Conclusion

The proposed algorithmic problem solving follows a new, universal decomposition method and is reproducible in *Mathematica* for any abstract operator equation of the kind of (3), with appropriate adjustments concerning operators and functions.

Define the operators of the initial problem

```
inverseAo = ∫_0^s #1 ⅆ#2 + (4*t)/3 * ∫_0^1 ∫_0^1 s*(∫_0^s #1 ⅆ#2) ⅆt ⅆs & (*integrate with respect to s1*)
inverseA = ∫_0^t #1 ⅆ#2 + (3*s^2)/2 * ∫_0^1 ∫_0^1 ∫_0^t #1 ⅆ#2 ⅆt ⅆs & (*integrate with respect to t1*)
f[t_, s_] := 5*t^2 + s; So[t_, s_] = t^3*s; Go[t_, s_] = t*s^2;
F = ∫_0^1 ∫_0^1 s^2*# ⅆt ⅆs &; Φ = ∫_0^1 ∫_0^1 t*# ⅆt ⅆs &;
```

Check correctness criterion for Bo

```
Lo = Det[IdentityMatrix[1] - Φ[inverseAo[Go[t, s1], s1]]]
```

Compute operator G

```
G[t_, s_] := inverseAo[So[t, s1], s1] +
  inverseAo[Go[t, s1], s1] * Lo^(-1) * Φ[inverseAo[So[t, s1], s1]]
```

Check correctness criterion for B

```
L = Det[IdentityMatrix[1] - F[G[t, s]]]
```

Compute the exact solution

```
inverseAinverseAo = inverseA[inverseAo[#1, #2] /. t → t1, t1] &
inverseAinverseAo[f[t, s1], s1] +
 inverseA[G[t, s] /. t → t1, t1] * L^(-1) * F[inverseAo[f[t, s1], s1]] +
 (inverseAinverseAo[Go[t, s1], s1] +
    inverseA[G[t, s] /. t → t1, t1] * L^(-1) * F[inverseAo[Go[t, s1], s1]]) *
  Lo^(-1) * Φ[inverseAo[f[t, s1], s1]]
```

FIGURE 2. *Mathematica* codes for solving Example 2

References

[1] N.N. Vassiliev, I.N. Parasidis, E. Providas. Exact solution method for Fredholm integro-differential equations with multipoint and integral boundary conditions. Part 2. Decomposition-extension method for squared operators. *Information and Control Systems* 2 (2019), pp. 2–9.

[2] I.N. Parasidis, E. Providas. Exact solutions to problems with perturbed differential and boundary operators. In: *Analysis and Operator Theory.* Springer Optimization and Its Applications, 146, pp. 301-317. Springer, Cham (2019).

[3] E. Providas, I.N. Parasidis. On the solution of some higher-order integro-differential equations of special form. *Vestnik SamU. Estestvenno-Nauchnaya Ser. (Vestnik of Samara University. Natural Science Series)*, 26:1 (2020), pp. 14–22.

[4] E. Providas, I.N. Parasidis. Extension operator method for solving nonstandard partial boundary value problems. *Kaz. Math. J.* 20:2 (2020) pp. 13-29.

[5] K.D. Tsilika. An exact solution method for Fredholm integro-differential equations. *Information and Control Systems* 4 (2019), pp. 2–8.

[6] Wolfram Research, Inc. *Mathematica, Version 11.3*, Champaign, Illinois, 2018.

K.D. Tsilika
Dept. of Economics,
University of Thessaly
Volos, Greece
e-mail: `ktsilika@uth.gr`