

УДК 004.4

Р.Я. Грудецький

Луцький національний технічний університет

**ВЗАЄМОДІЯ DELPHI-ДОДАТКІВ ТА ПРОГРАМНОГО СЕРЕДОВИЩА MATLAB**

*В статті розглядаються способи реалізації програмних інтерфейсів між Delphi-додатками і засобами системи Matlab на основі використання математичних DLL-бібліотек Matlab, вбудованого механізму Matlab Engine, стандартних засобів Windows для підтримки технології автоматизації.*

*Ключові слова:* програма, Delphi, Matlab, міжпрограмний інтерфейс, автоматизація.

*Літ. 4*

**Вступ.** Розробка користувацьких додатків для вирішення завдань аналізу та синтезу систем автоматичного управління, статистичної та спектральної обробки інформації, моделювання поведінки динамічних систем і т.п. практично завжди пов'язана з реалізацією складних математичних алгоритмів. Це векторно-матричний і поліноміальний аналіз, рішення систем звичайних диференціальних рівнянь, спектральний і кореляційний аналіз. Програмна реалізація таких алгоритмів вимагає наявності спеціалізованих бібліотек процедур і функцій з надійної та ефективної чисельної математики. У стандартних системах програмування цих коштів немає.

**В даний час існує декілька** професійних систем комп'ютерної математики, таких як MathCAD, Matlab тощо, можливості яких забезпечують ефективне вирішення зазначених вище завдань. Однак використання цих систем вимагає від користувача, крім доброго розуміння власне предметної області, вельми глибоких знань, як самої системи, так і наявності необхідного досвіду роботи в ній. Звичайному користувачеві, в цьому сенсі, вельми непросто реалізовувати складні обчислювальні проекти. Слід також зауважити, що в даний час професійні програми прикладного характеру повинні мати розвинений графічний інтерфейс з усіма атрибутами дружності - контролем введення даних, використанням відповідної термінології, вбудованою системою допомоги і т. п. Тому виникає актуальне **завдання** розробки додатків зі складної математичної функціональністю і з предметно-орієнтованим дружнім графічним інтерфейсом.

Для розробки додатків з професійним графічним інтерфейсом є цілком адекватні програмні інструментальні засоби – системи візуального програмування, наприклад, Delphi. Однак ці системи програмування у своєму розпорядженні мають досить обмежений набір процедур математичного характеру. Даний недолік систем програмування можна нівелювати, використовуючи можливості спеціалізованих математичних пакетів.

Розробниками системи Matlab пропонуються кілька варіантів використання засобів Matlab в автономних програмах або, навпаки, розширення можливостей Matlab об'єктами, створеними в системах програмування. Реалізовані і документовані засоби організації зв'язку з Matlab для додатків, написаних на мовах Fortran, C/C++, Java, C#. Є і відповідна література з цього питання [1, 2]. Проте в ній в основному переказується зміст фірмової документації, організація ж взаємодії Delphi і Matlab у зазначеному вище контексті не описана.

**Мета статті:** розглянути кілька способів організації використання Delphi-додатками засобів системи Matlab: звернення до DLL-бібліотек з математичними функціями ядра Matlab; застосування механізму Matlab Engine; реалізація технології Автоматизації стандартними засобами операційної системи Windows.

У загальному вигляді наведені вище способи організації міжпрограмних інтерфейсів, описати досить складно. Простіше проілюструвати запропоновані способи окремими простими прикладами Delphi-додатків.

Як відомо, Matlab підтримує роботу з декількома основними типами даних – числовими масивами, рядками, осередками, структурами [3]. Типом даних за замовчуванням є двовимірний числовий масив (речовинний або комплексний масив з елементами типу Double). Практично всі математичні функції Matlab працюють з цим типом даних. Тому в тестових Delphi-прикладних розглянута робота виключно з даними типу Double. Використання інших типів даних не викликає додаткових труднощів.

Призначення тестового Delphi-додатки полягає у виконанні таких операцій:

- а) введення користувачем вихідних числових даних у контексті мови Object Pascal;
- б) перетворення Pascal-даних до числовому типу Matlab-даних;
- в) передача перетворених даних того чи іншому засобу Matlab для виконання вибраних математичних операцій над цими даними;

- г) повернення результату математичної операції в Delphi-додаток;
- д) зворотне перетворення Matlab-даних у відповідні їм типам Pascal-даних;
- е) візуалізація засобами Delphi результатів математичних розрахунків в числовій або графічній формі.

Виконання першого і останнього пунктів не викликає особливих питань. Організація механізму виконання інших пунктів являє собою власне інтерфейс між Delphi-додатком і системою Matlab. Для підтримки цього міжпрограмного інтерфейсу в Delphi-додатку повинні бути сформовані адекватні мовні об'єкти – типи даних, змінні, функції і процедури. Всі дані об'єкти зручно розташувати в єдиному модулі (unit) і в міру необхідності наповнювати його новими програмними об'єктами. Назвемо цей модуль модулем сполучення засобів Matlab та системи програмування Delphi.

Для підготовки числових масивів для Matlab в цьому модулі необхідно оголосити такі типи даних:

```
type  
pDouble = A Double;  
mxComplexity = (m xREAL, mxCOMPLEX);  
pmxArray = A mxArray;  
mxArray = record  
end;
```

Тут pDouble - типізований покажчик на речовинний 8-байтовий тип Double; mxComplexity = (mxREAL, mxCOMPLEX) - перелічувальний тип для визначення речового або комплексного характеру числового масиву; mxArray - тип-запис для позначення особливої (порожньої) структури даних, що описує числовий масив; pmxArray – покажчик на запис mxArray.

Основне навантаження тут несе тип-запис mxArray. У нашому випадку цей тип досить позначити у вигляді порожнього запису. Змінні даного типу створюються спеціальними функціями з DLL-бібліотек Matlab [1]. При створенні такої змінної, поля запису автоматично заповнюються необхідною інформацією та звернення до них також забезпечують спеціальні функції DLL-бібліотек Matlab.

Всі версії системи Matlab аж до останньої мають у своєму складі особливу DLL-бібліотеку libmx.dll, яка містить C-функції, призначені для маніпуляцій з Matlab-даними – створення і видалення масивів даних, доступ до елементів масивів і до їх атрибутів. Призначення і короткі описи основних функцій цієї бібліотеки наведені в літературі [1, 2]. Для звернення до цих функцій з Delphi-додатків в модулі сполучення необхідно описати їх заголовки в стилі мови Object Pascal. Наприклад, функція, що виділяє необхідний обсяг пам'яті під двовимірний масив комплексних чисел, буде мати в модулі сполучення наступний заголовок:

```
function mxCreateDoubleMatrix (m, n: Integer; mData: mxComplexity): pmxArray;  
cdecl; external 'lib mx.dll';
```

У модуль сполучення слід додати ще кілька функцій/процедур, необхідних для маніпуляцій з числовими Matlab-масивами і матимуть наступні заголовки:

```
procedure mxSetName (arr_ptr: pmxArray; const name: String); cdecl; external 'libmx.dll';  
function mxGetPr (arr_ptr: pmxArray): pDouble; cdecl; external 'libmx.dll';  
function mxGetPi (arr_ptr: pmxArray): pDouble; cdecl; external 'libmx.dll';  
procedure mxDestroyArray (arr_ptr: pmxArray); cdecl; external 'libmx.dll';  
procedure mxFree (var ram: Pointer); cdecl; external 'libmx.dll';  
function mxTranspose (arr_ptr: pmxArray): pmxArray; cdecl; external 'libmx.dll';
```

Призначення цих функцій визначається їх назвами – присвоєння імені масиву даних, отримання покажчиків на речову та уявну частини масиву даних, знищення масиву і звільнення пам'яті, транспонування масиву.

Описи всіх заголовків повинні містити директиву cdecl (або stdcall) для узгодження різних способів обробки параметрів функцій, властивих мовам програмування Pascal і C.

Опишемо порядок дій при виконанні пунктів б) і д) - перетворення Pascal-даних в Matlab-дані і назад - Matlab-даних в Pascal-дані.

Розглянемо виконання прямого перетворення Pascal - Matlab. Припустимо, що мається двовимірний  $n \times m$  комплекснозначний масив  $Z$ , дійсна частина якого міститься в масиві  $X$ , а уявна – у масиві  $Y$ , тобто  $Z=X+iY$ . Для зручності подальших дій переформатуємо двовірні масиви  $X$  і  $Y$  по рядках в одноірні масиви. Потім виконаємо наступні дії:

- виділяємо пам'ять під  $n \times m$  комплекснозначний масив та іменуємо його як  $Z$

```
arrZ_ptr: = mxCreateDoubleMatrix (n, m, mxCOMPLEX);
mxSetName (arrZ_ptr, 'Z');
```

- копіюємо побайтно елементи дійсних ( $X$ ) і уявних ( $Y$ ) елементів масиву у відведену частину пам'яті для масиву  $Z$

```
Move (X [0], mxGetPr (arrZ_ptr)^, n*m*sizeof (Double));
Move (Y [0], mxGetPi (arrZ_ptr)^, n*m*sizeof (Double));
```

- транспондуєм двовимірний масив  $Z$ , так як операції з ним будуть проводити функції Matlab, які працюють з двовимірними масивами в стилі мови C,

```
arrZ_ptr: = mxTranspose (arrZ_ptr);
```

Тепер комплекснозначний масив  $Z$  (два дійсних Pascal-масиву  $X$  і  $Y$ ) конвертований в структуру даних, коректно сприйняту системою Matlab як звичайний масив комплексних чисел. Показчик на цю структуру міститься у змінній  $arrZ\_ptr$ .

Зворотне перетворення Matlab - Pascal виконується таким чином. Припустимо, що мається одновимірний  $1 \times n$  числовий Matlab-масив, на який вказує  $arrM\_ptr$ . У Delphi-додатку для його прийому приготований звичайний одновимірний  $1 \times n$  числовий масив  $Mas$ , наприклад, визначений як `var Mas: array of Double`. Операція зворотної конвертації здійснюється, наприклад, такою послідовністю Pascal-інструкцій:

```
SetLength (Mas, n);
Move (mxGetPr (arrM_ptr)^, Mas [0], n*sizeof (Double));
```

Підготовлені у форматі `mxArray` числові масиви можуть бути передані тим чи іншим засобам Matlab для виконання математичних операцій над ними. Розглянемо коротко деякі способи звернення до цих засобів з додатків Delphi.

#### 1. Використання математичної бібліотеки C/C++

Аж до сьомої версії математичні процедури загального призначення системи Matlab були оформлені у вигляді математичної бібліотеки C/C++. Починаючи з версії Matlab 7.0, математична бібліотека C/C++ була замінена на середовище виконання компонентів Matlab MCR (Matlab Component Runtime).

Математична бібліотека Matlab 6.x включає в себе більше 400 математичних процедур і реалізована у вигляді набору DLL-бібліотек з 48-ти файлів. Для вільного розповсюдження та використання всі файли бібліотеки упаковані в архів `mglnstaller.exe`. Процедури математичної бібліотеки написані на мовах C і C++. Тому для використання їх у додатках Delphi необхідно коректним чином прописати заголовки необхідних процедур в модулі сполучення. Імена всіх процедур у бібліотеці починаються із префікса `mlf`. Наприклад, процедура обчислення власних векторів і власних чисел матриці матиме в Delphi-модулі наступний заголовок:

```
function mlfEig (D, A, B: pmxArray): pmxArray; cdecl; external 'libmatlb.dll';
```

Тут вказано, що тіло функції `mlfEig` розташоване у зовнішній (external) DLL-бібліотеці `libmatlb.dll`.

Практично весь набір функцій математичної бібліотеки Matlab описаний в контексті мови C у фірмовій документації [4]. Коректне уявлення заголовків C-функцій в термінах мови Pascal, з

урахуванням описаних вище типів даних, дозволяє використовувати в додатках Delphi всю міць ядра математики Matlab.

Досвід застосування даного механізму організації зв'язку Delphi-додатків і ядра математики Matlab довів свою ефективність в безлічі конкретних програм. Однак цей механізм не дозволяє використовувати математичні засоби Matlab, розташовані в його спеціалізованих пакетах – Toolboxes. Досить часто саме там і знаходяться найбільш цікаві та цінні математичні процедури, необхідні для вирішення конкретної спеціальної завдання. Наступний механізм дозволяє ліквідувати цей недолік.

## 2. Використання технології Matlab Engine

Дана технологія, запропонована самими розробниками Matlab, дозволяє отримати програмний доступ не тільки до математичного ядру Matlab, але і практично до всіх його спеціалізованих пакетів. Правда за це доводиться платити велику ціну – передбачається повний доступ до систем Matlab. Для індивідуального користувача це означає наявність встановленого Matlab в його комп'ютерній системі.

Суть даної технології полягає в тому, що додаток, коли в цьому виникає необхідність, ініціює (завантажує) Matlab і управляє його роботою - передає в робочу область дані, пересилає в командне вікно необхідні команди, повертає в додаток дані з робочої області, деініціалізує (вивантажує) Matlab.

Технологія Matlab Engine являє собою одне з розширень загальної технології COM, а саме - Автоматизацію. У даному випадку Matlab виступає в ролі сервера автоматизації, а Delphi - додаток - у ролі диспетчера (контролера) автоматизації. Інтерфейси COM-об'єктів обгорнуті в декілька C-функцій, які організовані в DLL-бібліотеку libeng.dll.

Для доступу програми Delphi до Matlab Engine необхідно в Pascal-модуль сполучення додати опису ще двох специфічних типів даних:

```
type  
Engine = record end;  
pEngine = ^Engine;
```

і, як мінімум, п'яти функцій з бібліотеки libeng.dll:

```
function engOpen (par: pointer): pEngine; cdecl; external 'libeng.dll';  
function engClose (par: pEngine): pEngine; cdecl; external 'libeng.dll';  
function engGetVariable (ep: pEngine; const var_name: PChar) : pmxArray; cdecl; external 'libeng.dll';  
function engPutVariable (ep: pEngine; st: PChar; data: pmxArray): pEngine; cdecl; external 'libeng.dll';  
function engEvalString (ep: pEngine; const parst: PChar): pEngine; cdecl; external 'libeng.dll';
```

Показчик на пустий рядок pEngine ідентифікує завантажену додатком копію Matlab. Перші дві функції застосовуються для завантаження і вивантаження Matlab; наступні дві для обміну даними між додатком і робочою областю Matlab; остання функція служить для передачі в командне вікно Matlab командного рядка на виконання операції.

Звернення з Delphi-додатки до Matlab (передача рядка-команди) імітує роботу користувача з системою Matlab в режимі суперкалькулятора (в режимі командного рядка). Передані в Matlab і одержувані Delphi-додатком з Matlab'a дані упаковані в структури типу mxArray. Pascal-модуль сполучення, що забезпечує механізм Matlab Engine, повинен містити виключно функції з DLL-бібліотек libmx.dll і libeng.dll. Ці бібліотеки входять до складу всіх версій Matlab.

Технологія Matlab Engine є фірмовою (MathWorks) оболонкою механізму Automation, підтримуваного операційною системою Windows. Природно, виникає бажання безпосередньо реалізувати цей системний механізм для організації міжпрограмного інтерфейсу між Delphi-додатком-клієнтом і Matlab-сервером. Система програмування Delphi має у своєму складі засоби організації такого механізму. Цей факт і дозволяє реалізувати в "чистому вигляді" технологію Автоматизації.

## 3. Використання системи Matlab як сервера Автоматизації

Для роботи з Matlab як з сервером Автоматизації необхідно знати властивості та методи (інтерфейси), які надає сервер Автоматизації. Ця інформація доступна з фірмової документації [4].

Обмін даними між Delphi-додатком і Matlab з боку Delphi-додатки (диспетчера Автоматизації) здійснюється за допомогою змінних варіантного типу (тип OLEVariant). Matlab, як  
© Грудецький Р.Я.

сервер Автоматизації, надає Delphi-додатком свої функції PutFullMatrix і GetFullMatrix для обміну числовими масивами, а також функцію Execute для передачі команди на виконання операції сервером Автоматизації. Функції CreateOleObject і Quit відповідно відкривають і закривають сервер (Matlab).

Як приклад даного способу організації зв'язку Delphi-додатки з Matlab наведемо фрагменти Delphi-коду для вирішення задачі обчислення рангу числової  $n \times m$ -матриці.

Визначимо необхідні змінні:

```
var
Mtlb: Variant;
A: array of array of Double;
A_re, A_im, R_re, R_im: OLEVariant;
```

Змінні призначені для ідентифікації сервера Автоматизації (Mtlb), зберігання вихідної двовимірної матриці (A), зберігання числової (A\_re) і уявної (A\_im) частин матриці A, зберігання числової (R\_re) і уявної (R\_im) частин значення рангу матриці A відповідно.

Виконувани оператори фрагмента коду:

```
{1}      SetLength (A, n, m);
{2}      for i: = 1 to n do
{3} for j: = 1 to m do A [i-1, j-1]: = random;
{4}      A_re: = VarArrayCreate ([0, n-1, 0, m-1], varDouble);
{5}      A_im: = VarArrayCreate ([0, N-1, 0, M-1], varDouble);
{6}      for i: = 0 to n-1 do
{7} for j: = 0 to m-1 do
{8} begin A_re [i, j]: = A [i, j]; A_im [i, j]: = 0 end;
{9} Mtlb: = CreateOleObject ('Matlab.Application');
{10} Mtlb.PutFullMatrix ('A', 'base', VarArrayRef (A_re), VarArrayRef (A_im));
{11} Mtlb.Execute ('r = rank (A);');
{12} R_re: = VarArrayCreate ([1,1], varDouble);
{13} R_im: = VarArrayCreate ([1,1], varDouble);
{14} Mtlb.GetFullMatrix ('r', 'base', VarArrayRef (R_re), VarArrayRef (R_im));
{15} Mtlb.Quit;
```

Тут у рядках 1-3 виділяється пам'ять під масив A і генеруються значення елементів масиву A. Інструкції у рядках 4-8 формують і заповнюють два двовимірних речових варіантних масиву A\_re і A\_im під речову і уявну частини матриці A. У рядку 9 диспетчер Автоматизації (Delphi-додаток) ініціює сервер Автоматизації (Matlab). Оператори в рядках 10 і 11 призначені відповідно для передачі масиву A в робочу область і виконання команди обчислення рангу матриці A. Інструкції 12-ого і 13-ого рядків готують змінну варіантного типу для прийому значення рангу матриці з робочої області Matlab. У 14-ій рядку відбувається передача значення змінної Matlab r (рангу матриці) в варіантну змінну (R\_re, R\_im) програми Delphi. Інструкція у рядку 15 закриває сервер Автоматизації.

**Висновки.** Розглянуті вище способи організації міжпрограмного інтерфейсу Delphi-додатків і засобів системи Matlab були неодноразово реалізовані в практичній роботі і показали свою надійність і ефективність.

Подальший напрямок роботи - використання останніх версій пакету Matlab Compiler для створення користувацьких DLL-бібліотек з m-функцій і підключення їх до додатків Delphi.

1. Подкур М.Л. Программирование в среде Borland C++ Builder с математическими библиотеками Matlab C/C++ / М.Л. Подкур, П.Н. Подкур, Н.К. Смоленцев. — М.: ДМК Пресс, 2006. — 496 с.

2. Смоленцев Н.К. MATLAB и программирование на Visual C#, Borland JBuilder, VBA: Учебный курс (+CD) / Н.К. Смоленцев. — М.: ДМК Пресс; СПб.: Питер, 2009. — 464 с.

3. MATLAB 7. Programming Fundamentals. - MathWorks, Inc., 2010. - 684 p.

4. MATLAB 7. External Interfaces. - MathWorks, Inc., 2010. - 776 p.