УДК 681.3.06

А.М. Ковальчук, аспір.
Житомирський інженерно-технологічний інститут

ВВЕДЕННЯ ТИПІЗАЦІЇ БАЗИСНИХ ФУНКЦІЙ В МЕТОДАХ АПРОКСИМАЦІЇ ФУНКЦІОНАЛЬНИХ ВІДНОШЕНЬ

(Представлено к.т.н. доц. М.М. Колодницьким)

Запропонована методика визначення та використання типізованих базисних функцій в методах апроксимації функціональних відношень. Проведено аналіз степеневих, тригонометричних та експоненціальних базисних функцій. Наведено приклади програмної реалізації систем базисних функцій та результати роботи програмної системи, яка використовує бібліотеку алгоритмів, що створена з використанням типізованих базисних функцій.

Як відомо, апроксимацією функції називають заміну за деяким правилом функції $f(t)$ близькою до неї, у тому чи іншому розумінні, функцією $\psi(x)$ із раніше вибраної множини Q [1]. При цьому, однією з головних задач апроксимації є задача вибору множини Q базисних функцій $\{\varphi_i(x)\}$, якими наближають функцію $f(x)$. Часто на практиці розглядають випадок апроксимації функції $f(t)$ за допомогою поліному (багаточлену):

$$\psi(x) = a_0 \varphi_0(x) + a_1 \varphi_1(x) + \ldots + a_m \varphi_m(x), \qquad (1)$$

де базисом є деяка система лінійно-незалежних функцій $\{\varphi_1(x), \varphi_2(x), \ldots, \varphi_m(x)\}$, яку можна вибирати в залежності від умов конкретної задачі та апріорних вимог до $\psi(x)$. Зокрема, $\varphi_k(x)$ можуть бути степеневими, тригонометричними, експоненціальними, дробовими раціональними функціями тощо [2].

Розглянемо класичну задачу лінійної інтерполяції таблично заданої функції. Нехай деяка функція $y = f(x)$ задана рядом значень $y_0 = f(x_0)$, $y_1 = f(x_1)$, ..., $y_n = f(x_n)$, які вона набуває у вузлах деякої сітки $\Delta = \{x_0, x_1, \ldots, x_n\}$.

Виберемо систему базисних функцій $\{\varphi_k(x)\}$ на відрізку $[x_0, x_n]$, слід зазначити, що функції $\varphi_k(x)$ мають бути лінійно незалежними. Функцію $f(x)$ замінюємо функцією $\psi(x)$, що утворена, як лінійна комбінація базисних функцій $\varphi_k(x)$:

$$\psi(x) = \sum_{k=0}^{m} a_k \cdot \varphi_k(x), \qquad (2)$$

де a_k – деякі невідомі коефіцієнти розкладу. Для визначення цих коефіцієнтів потрібно мати $m+1$ рівнянь, тобто утворити систему з $m+1$ лінійних алгебраїчних рівнянь (СЛАР) виду:

$$A \cdot \bar{x} = \bar{b}, \qquad (3)$$

де вектором невідомих $\bar{x}$ виступають параметри a_k, тобто:

$$\bar{x} = \{a_0, a_1, \ldots, a_m\}.$$

Для того, щоб ця система рівнянь мала єдиний розв'язок, необхідно, щоб рівняння в (2) були лінійно незалежними. Ця вимога, в свою чергу, перетворюється у вимогу лінійної незалежності системи базисних функцій $\varphi_k(x)$ [3].

Для визначення параметрів a_k лінійного розкладу (1) необхідно утворити систему лінійних рівнянь (3). В основу способу утворення системи рівнянь покладено умову співпадання значень початкової функції $f(x)$ та функції $\psi(x)$, взятих у вузлах сітки Δ, тобто взятих для кожного окремого значення незалежної змінної $f(x_i) = \psi(x_i)$. Використання цієї умови дає шукану СЛАР виду (3):

$$\begin{cases} \varphi_0(x_0)a_0 + \varphi_1(x_0)a_1 + \ldots + \varphi_m(x_0)a_m = f(x_0) \\ \varphi_0(x_1)a_0 + \varphi_1(x_1)a_1 + \ldots + \varphi_m(x_1)a_m = f(x_1) \\ \ldots\ldots\ldots\ldots\ldots \\ \varphi_0(x_n)a_0 + \varphi_1(x_n)a_1 + \ldots + \varphi_m(x_n)a_m = f(x_n) \end{cases} \qquad (4)$$

Як видно з виразу (4), для утворення такої системи ми повинні вирахувати значення базисних функцій в точках сітки Δ. Потрібно звернути увагу на те, що при переході до іншої системи базисних функцій метод формування СЛАР не змінюється. Тому, була створена методика за якою виділяються спільні параметри базисних функцій, та визначається шаблон для опису базисних функцій. Розглянемо детально процедуру обчислення для кожного з найбільш відомих типів базисних функцій. Слід зазначити, що для формування СЛАР доцільно використовувати покрокову рекурентну схему обчислення базисних функцій в точках сітки, тому що це суттєво зменшує об'єм обчислень.

Степеневі функції. Система базисних функцій має вигляд:

$$\varphi_k(x) = x^k,$$

для $k = 0 \dots m$ або у рекурентному вигляді:

$$\varphi_k(x) = \begin{cases} 1 & k = 0; \\ \varphi_{k-1}(x) \cdot x & k > 0. \end{cases}$$

Тригонометричні функції:

$$\varphi_k(x) = \begin{cases} 1 & k = 0; \\ sin\,(k \cdot x) & k - \text{непарне}; \\ cos\,(k \cdot x) & k - \text{парне}, \end{cases}$$

для $k = 0 \dots m$, у рекурентному вигляді система визначається так само.

Експоненціальні функції. Система має вигляд:

$$\varphi_k(x) = e^{k \cdot x},$$

для $k = 0 \dots m$ або у рекурентному вигляді:

$$\varphi_k(x) = \begin{cases} 1 & k = 0; \\ \varphi_{k-1}(x) \cdot e^x & k > 0. \end{cases}$$

У створеній бібліотеці методів різні типи базисних функцій програмно реалізовані за однією схемою, що дає нам можливість використовувати їх в методах лінійної інтерполяції, без адаптації іншого програмного коду під конкретний тип базисних функцій. Далі наведемо приклад програмної реалізації систем базисних функцій, рис. 1.

```
typedef CDSRReal (*BASEFUNC)
        ( long it, unsigned long size, CDSRReal arg, CDSRReal *pPrev );

// Степеневі поліноми
CDSRReal pol_lagr( long it, unsigned long size, CDSRReal arg, CDSRReal *pPrev )
{
        return it == 0 ? 1.0 : pPrev[ it - 1 ] * arg;
}

// Тригонометричні поліноми
CDSRReal pol_trgn( long it, unsigned long size, CDSRReal arg, CDSRReal *pPrev )
{
        if ( size % 2 == 0 )
                throw EMID_NUMBER_OF_ELEMENTS_MUST_BE_ODD;

        return it == 0 ? 1.0 :
                it % 2 ? cos(((_it + 1) >> 1) * arg) : sin(( _it >> 1) * arg);
}

// Експоненціальні поліноми
CDSRReal pol_exp( long it, unsigned long size, CDSRReal arg, CDSRReal *pPrev )
{
        return it == 0 ? 1.0 :
                it == 1 ? exp((CDSRReal) arg) : pPrev[ it - 1 ] * pPrev[ 1 ];
}

// Поліноми Чебишева
CDSRReal pol_cheb( long it, unsigned long size, CDSRReal arg, CDSRReal *pPrev )
{
        if ( it == 0 )
                return 1;
        else if ( it == 1 )
                return arg;
        else
                return 2.0 * arg * pPrev[ _it - 1 ] - pPrev[ _it - 2 ];
        }
}
```

Рис. 1. Приклад програмної реалізації систем базисних функцій

Наведемо приклад програмної реалізації методу побудови СЛАР, для визначення коефіцієнтів апроксимації.

```
void _getApprPolinom(
        CDSRMVector<CDSRReal> *pRealArg,
        CDSRMVector<CDSRReal> *pRealFunc,
        CDSRArray<CDSRReal> *pPol,
        ALMAFUNC_APPRBASIS type )
{
        if ( pRealArg->size() != pRealFunc->size() )
                throw EMID_SIZE_OF_VECTORS_NOT_EQUAL;

        unsigned long i, j;

        CDSRMMatrix<CDSRReal> matrix( pRealArg->size(), pRealArg->size() );
        for ( i = 0; i < pRealArg->size(); i++ )
                for ( j = 0; j < pRealArg->size(); j++ )
                        matrix( i, j ) = (*pBaseFunc[ type ])
                                ( j, pRealArg->size(), (*pRealArg)[ i ],
                                        matrix.begin() + i * matrix.n_column() );

        _LUCR( &matrix, pRealFunc, & pPol );
}
```

Рис. 2. Приклад програмної реалізації методу побудови СЛАР

Для випадку інтерполяції за Ермітом, тобто якщо при створенні СЛАР для знаходження коефіцієнтів апроксимуючого поліному використовуються похідні першого, другого та вищого порядків, створюється набір, аналогічний функції, до якого додасться параметр, що відповідає за номер похідної.

На рис. 3–8, наведено приклад роботи програмної системи "DSR Open Lab 1.0" [4, 5], у якій використовується розроблена бібліотека алгоритмів апроксимації функцій.

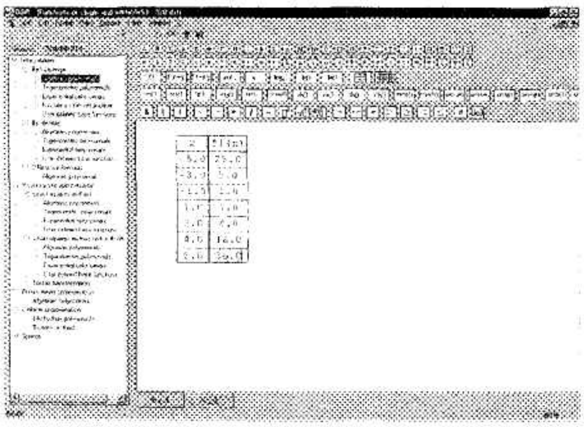

Рис. 3. Дані для апроксимації

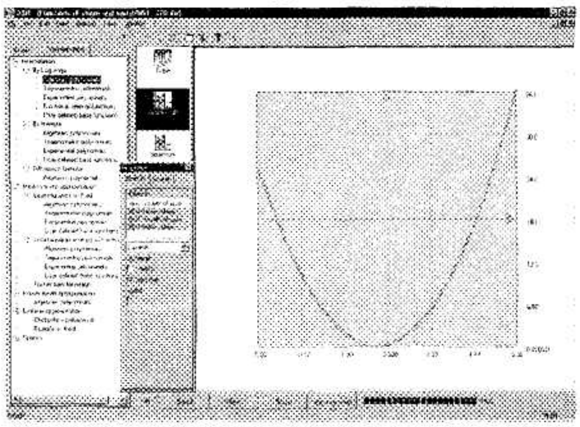

Рис. 4. Результати апроксимації

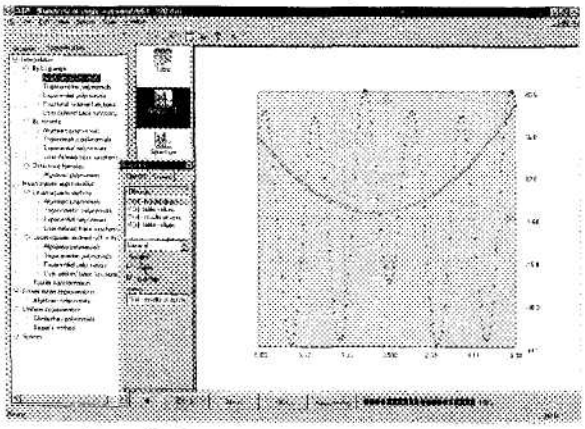

Рис. 5. Результати апроксимації

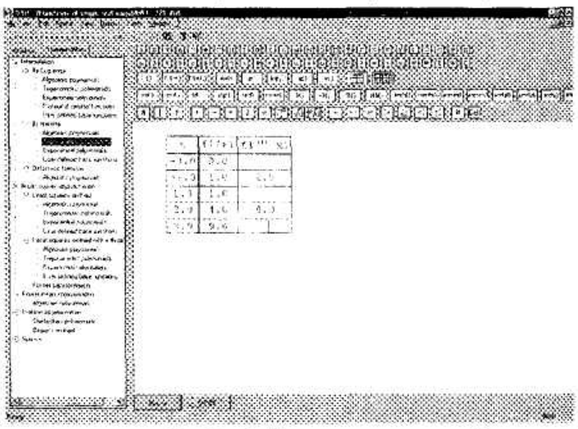

Рис. 6. Дані для апроксимації

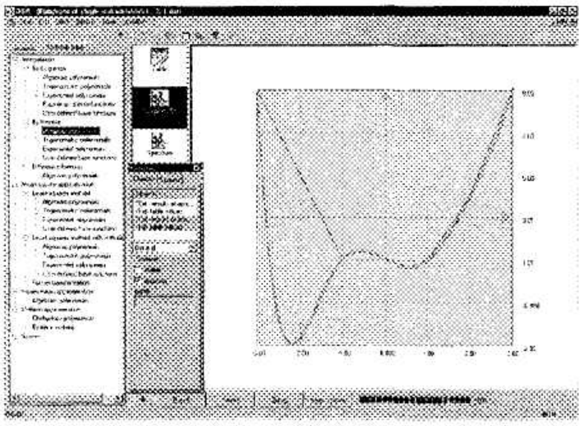

Рис. 7. Результати апроксимації

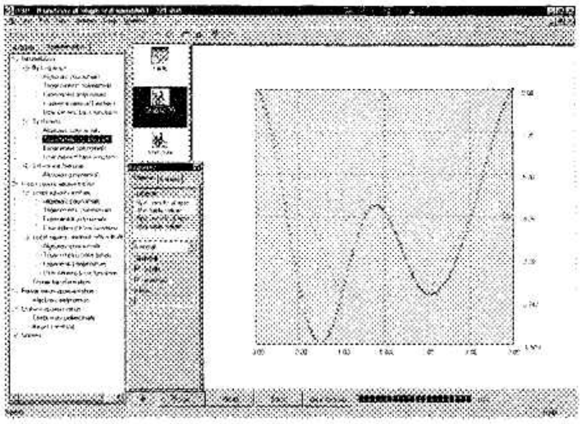

Рис. 8. Результати апроксимації

Висновки

Використовуючи бібліотеку алгоритмів з апроксимації функціональних відношень у програмному комплексі "DSR Open Lab 1.0", було проведено ряд тестів. Визначено, що методи, які створені на основі запропонованої методики, є досить точними та швидкодіючими. Зазначимо також, що запропонована методика визначення та використання типізованих базисних функцій дає змогу суттєво зменшити об'єм програмного коду для розробки бібліотеки алгоритмів з апроксимації функціональних відношень.

ЛІТЕРАТУРА:

1. *Колодницький М.М.* Основи теорії математичного моделювання систем. – Житомир, 2000 (в друці).
2. *David M. Yang, Robert Todd Gregory* A Survey of Numerical Mathematics, Dover Publications Inc., New York, 2000.
3. *Хемминг Р.В.*, Численные методы. – М.: Наука, 1972.
4. *Kolodnytsky M., Kovalchuk A., Kuryata S., Levitsky V.* "A Theory of Approximation of Functions" Courseware // Proceedings of the 5th Asian Technology Conference in Mathematics, December 17–21, Chiang Mai, Thailand, 2000. – P. 86–95.
5. *Kolodnytsky M., Ivanitsky I., Kovalchuk A., Kuryata S., Levitsky V.* "DSR Open Lab 1.0". – software system for simulation // The 21st International Conference on Information Technology Interfaces, Pula, Chroatia, 1999. – 31 p.

КОВАЛЬЧУК Андрій Михайлович – аспірант Житомирського інженерно-технологічного інституту.

Наукові інтереси:

- комп'ютерні інформаційні технології;
- моделювання і розв'язок задач за допомогою обчислювальної техніки;
- використання обчислювальної техніки в навчальному процесі;
- комп'ютерна графіка.

Подано 14.12.2000