УДК 004.777:37.01

Малашонок Г. І.

ХМАРНА МАТЕМАТИКА MATHPARTNER У КИЄВО-МОГИЛЯНСЬКІЙ АКАДЕМІЇ

У статті подано короткий опис сервісу MathPartner (його встановлено на сервері Києво-Могилянської академії mathpartner.ukma.edu.ua) – хмарної системи символьних обчислень – і тих можливостей, які вона надає для вдосконалення освітнього процесу у вищій школі. Описано застосування сервісу GitHub як сховища для навчальних матеріалів, а також нового засобу підготовки та проведення лекцій. Зроблено висновок про те, що застосування хмарної математики MathPartner сприяє інтенсифікації освіти у всіх сферах, де використовується математика. Крім того, хмарна математика є універсальним інструментом застосування математичних знань, що сприяє розвитку прикладної математики.

Ключові слова: хмарна математика, система комп'ютерної алгебри, MathPartner, символьні обчислення.

Вступ

В останні роки хмарні сервіси приходять на зміну прикладним пакетам програм. Це саме стосується і такої сфери, як системи символьних обчислень. В останнє десятиліття з'явилися такі SaaS системи символьних обчислень, як MathPartner, Sage, Wolfram Alpha і Wolfram Cloud.

Кількість користувачів систем символьних обчислень серед тих, хто вивчає і застосовує математику, досі була незначною. Ця ситуація швидко змінюється з появою SaaS систем і їх широким застосуванням. Можна очікувати, що хмарні математичні сервіси приведуть до кардинальної зміни всієї системи освіти, до зміни статусу математичного знання в сучасному суспільстві. Математичними знаннями можна буде ефективно користуватися, уникаючи при цьому складних викладок.

Сервіс MathPartner з'явився в 2011 році і був одним із перших серед хмарних систем [1]. Сьогодні цей сервіс впроваджений у Києво-Могилянській академії і доступний за адресою: mathpartner.ukma.edu.ua.

Для звернення до сервера не потрібно купувати додаткові програми, можна скористатися будь-яким сучасним браузером. Основним об'єктом на сервері є зошит користувача, він зберігає всі введені користувачем записи. Тексти в зошит можна завантажувати і вивантажувати як звичайні текстові файли або редагувати, як у звичайному редакторі.

Цю статтю присвячено опису MathPartner і тим новим можливостям, які він надає щонайширшому колу користувачів: від професіоналів математиків – до молодших школярів, від фізиків-теоретиків – до вчителів математики та фізики.

1. Бібліотека математичних текстів

На веб-сервісі GitHub розташована бібліотека текстів на мові Mathpar. Це мова, яка використовується на сервісі MathPartner. Вона доступна за адресою: http://github.com/mathpar.

Прямо на сайті можна переглядати текстові файли, але можна скачати і весь архів. Сервіс підтримує отримання і редагування коду через Git, SVN і Mercurial. Можна зареєструватися на веб-сервісі GitHub і після цього розміщувати свої тексти та отримувати тексти, розміщені іншими учасниками. Можна, крім того, завантажити архів усієї бібліотеки у вигляді zip-архіву: https://github.com/mathpar/mathpar/archive/master.zip. Адреса кореня архіву: https://github.com/mathpar/mathpar/tree/master/LIBRARY/mathpar. Наприклад, якщо вибрати шлях EUROPE / Ukraine-ua / UNIVERSITY / MATH / Computer_Algebra / Malaschonok / Computer_Algebra_Chap_01, то будуть доступні всі параграфи першого розділу підручника з комп'ютерної алгебри. Можна відкрити і читати будь-який з них. Щоб редагувати текст, у верхньому меню є олівець. Якщо в режимі редагування виконати Ctrl-A + Ctrl-C, то можна скопіювати весь текст, а потім зберегти його в себе в текстовому редакторі.

Ось так, наприклад, адресується перший параграф першого розділу підручника:

github. com / mathpar / mathpar / tree / master / LIBRARY / mathpar / EUROPE / Ukraine-ua / UNIVERSITY / MATH / Computer_Algebra / Malaschonok / Computer_Algebra_Chap_01 / 101_lect_CA.txt.

Нижче наведено фрагмент початку цього параграфа.

```
"$\bf\hbox {ТЕОРЕМА (Китайська теорема про залишки).}$
Нехай $R$ – комутативне кільце з одиницею, $m_1, ..., m_k$ – взаємно-прості елементи кільця,
$m = m_1m_2 ... m_k$ і задані довільні елементи $r_1, ..., r_n \in R$, тоді існує елемент $x \in
R$, єдиний по $\mod(m)$, такий, що виконуються рівності:
$$r_1 = x \mod(m_1), r_2 = x \mod(m_2), ..., r_n = x \mod(m_n).$$
$\bf\hbox{ДОВЕДЕННЯ (Єдиність).}$
Нехай знайдуться два розв'язки $x_1, x_2$. Знайдемо різницю $y = x_1-x_2$, тоді
$$0 = y \mod(m_1),  ..., 0 = y \mod(m_n)$$
Оскільки $m_1, ..., m_n$ взаємно-прості, то $НСК(m_1, ..., m_n) = m_1 m_2 ... m_n$. Отже, $y$
кратне $m = m_1 m_2 ... m_n$, тобто $0 = (x_1-x_2) mod(m)$.
Єдиність доведена.
$\bf\hbox{ДОВЕДЕННЯ (Достатність. Розв'язання методом Лагранжа).}$
Позначимо $M_i = m_1 ... m_{i-1} m_{i+1} ... m_n, $ (1=1,..,n).
Позначимо $N_i$ (1=1,..,n) елемент, обернений до $M_i$ за модулем $m_i$: $1 = N_i M_i
mod(m_i)$.
Розглянемо вираз: $a = r_1 M_1 N_1 + r_2 M_2 N_2 + .. + r_n M_n N_n$
Зауважимо, що:
$M_i N_i \mod(m_j) = 1$, якщо $i = j$, та $M_i N_i \mod(m_j) = 0$, якщо $i \ne j$.
Отже, $r_j = a \mod(m_j)$.
$\square$
$\bf\hbox{ПРИКЛАД}$:
Потрібно знайти найменше додатне число $a$ таке, що
$3 = a\mod(5)$, $2 = a\mod(8)$, $1 = a\mod(3)$.
$\bf\hbox {РОЗВ'ЯЗАННЯ методом Лагранжа}$"
SPACE = Z[];
"За умовою нам дано модулі: " m_1 = 5; m_2 = 8; m_3 = 3;
"і залишки: " r_1 = 3; r_2 = 2; r_3 = 1;
"Знайдемо добуток усіх модулів: " m = m_1 m_2 m_3;
"Знайдемо співмножники: " M_1 = m / m_1; M_2 = m / m_2; M_3 = m / m_3;
"Знайдемо обернені за модулем: " N_1 = \modInverse(M_1, m_1);
N_2 = \modInverse(M_2, m_2); N_3 = \modInverse(M_3, m_3);
"В результаті отримаємо:  "a = \mod(r_1 M_1 N_1 + r_2 M_2 N_2 + r_3 M_3 N_3, m)
```

Весь файл можна повністю зберегти, а потім завантажити у вікно робочого зошита на сервісі MathPartner. Але можна і за допомогою засобів copy-paste перенести тільки цей фрагмент.

Після виконання цього фрагмента буде виведено таке зображення, як на рисунку нижче. Зазначимо, що число 58, яке стоїть після мітки *out*, – це результат, обчислений при виконанні цієї програми, а текст програми, вхідні дані і коментарі розташовані вище.

2. Підготовка лекцій та інших математичних текстів

Звернемося до наведеного фрагмента тексту на мові Mathpar і порівняємо його з отриманим зображенням. Легко бачити прості правила, які використовуються для запису математичних виразів: 1) перед нижнім індексом стоїть знак «підкреслення»; 2) перед ступенем стоїть знак «шапочка»; 3) перед назвою функції стоїть обернена скісна риска; 4) оператор *SPACE = Z[x, y]* відповідає за вибір числової множини та назви змінних.

Для розмітки речень у тексті прийнято такі правила; 1) формули треба оточувати знаками долара; 2) якщо оточувати подвійними знаками доларами, то вираз розміститься в центрі рядка; 3) якщо всередині додати оточення

ТЕОРЕМА (Китайська теорема про залишки).
Нехай R – комутативне кільце з одиницею, $m_1, \ldots, m_k$ – взаємно-прості елементи кільця, $m = m_1 m_2 \ldots m_k$ і задані довільні елементи $r_1, \ldots, r_n \in R$, тоді існує елемент $x \in R$, єдиний по $\mathbf{mod}(m)$, такий, що виконуються рівності:

$$r_1 = \text{x}\,\mathbf{mod}(m_1), r_2 = \text{x}\,\mathbf{mod}(m_2), \ldots, r_n = \text{x}\,\mathbf{mod}(m_n).$$

ДОВЕДЕННЯ (Єдиність).
Нехай знайдуться два розв'язки x_1, x_2. Знайдемо різницю $y = x_1 - x_2$, тоді

$$0 = y\,\mathbf{mod}(m_1), \ldots, 0 = y\,\mathbf{mod}(m_n)$$

Оскільки $m_1, \ldots, m_n$ взаємно-прості, то $HCK(m_1, \ldots, m_n) = m_1 m_2 \ldots m_n$. Отже, y кратне $m = m_1 m_2 \ldots m_n$, тобто $0 = (x_1 - x_2) mod(m)$.
Єдиність доведена.
ДОВЕДЕННЯ (Достатність. Розв'язання методом Лагранжа).
Позначимо $M_i = m_1 \ldots m_{i-1} m_{i+1} \ldots m_n$, (1=1,..,n).
Позначимо N_i (1=1,..,n) елемент, обернений до M_i за модулем m_i: $1 = N_i M_i mod(m_i)$.
Розглянемо вираз: $a = r_1 M_1 N_1 + r_2 M_2 N_2 + .. + r_n M_n N_n$
Зауважимо, що:
$M_i N_i\,\mathbf{mod}(m_j) = 1$, якщо $i = j$, та $M_i N_i\,\mathbf{mod}(m_j) = 0$, якщо $i \neq j$.
Отже, $r_j = a\,\mathbf{mod}(m_j)$.
□
ПРИКЛАД:
Потрібно знайти найменше додатне число a таке, що
$3 = a\,\mathbf{mod}(5), 2 = a\,\mathbf{mod}(8), 1 = a\,\mathbf{mod}(3)$.
РОЗВ'ЯЗАННЯ методом Лагранжа
$SPACE = Z[];$
За умовою нам дано модулі:$m_1 = 5; m_2 = 8; m_3 = 3;$
і залишки:$r_1 = 3; r_2 = 2; r_3 = 1;$
Знайдемо добуток усіх модулів:$m = m_1 \cdot m_2 \cdot m_3;$
Знайдемо співмножники:$M_1 = m/m_1; M_2 = m/m_2; M_3 = m/m_3;$
Знайдемо обернені за модулем:$N_1 = \mathbf{modInverse}(M_1, m_1);$
$N_2 = \mathbf{modInverse}(M_2, m_2); N_3 = \mathbf{modInverse}(M_3, m_3);$
В результаті отримаємо: $a = \mathbf{mod}(r_1 M_1 N_1 + r_2 M_2 N_2 + r_3 M_3 N_3, m)$
out :

58

\bf\hbox{}, то текст буде написаний жирним шрифтом.

Текст у мові Mathpar, на відміну від операторів, потрібно брати в подвійні лапки. Оператори розділяються крапкою з комою або текстом.

Всі оператори виконуються послідовно, і результат останнього оператора з'являється після тексту. Якщо ж зустрічаються команди друку, то виводиться не останній оператор, а результати цих команд.

Таким чином, з'являється можливість не тільки викладати навчальний матеріал, а й супроводжувати його прикладами. Ці приклади стають відразу готовим інструментом, яким можна користуватися для практичного опанування нового матеріалу і самостійного розв'язання задач.

3. Читання лекцій

Розглянемо організацію лекції з використанням сервісу MathPartner. В аудиторії, де проходить лекція, має бути доступ до Інтернету. Потрібно вийти на сервіс за адресою: http://mathpartner.ukma.edu.ua і перейти в робочий зошит. У лівому полі відкрити меню **«Файли»** і вибрати верхній рядок **«Завантажити текст».** Відкриється шлях до файлів на комп'ютері, і потрібно вказати файл з текстом лекції. Цей файл завантажиться у вигляді послідовності вікон.

Над кожним вікном ліворуч є три керуючих знаки: **«трикутник»** – запустити виконання, **«перемикач»** – переключити режим зображення на режим текст, **«плюс»** – додати нижче нове порожнє вікно. Після запуску на виконання відбувається виведення результату, і режим текст змінюється на режим зображення. Щоб повернути режим редагування тексту, достатньо скористатися «перемикачем» або просто клікнути мишкою в потрібному вікні.

Лектор може послідовно демонструвати фрагменти лекції, розділені вікнами, послідовно натискаючи кнопки **«виконати»** і коментуючи отримані результати. Він може змінювати готові приклади на нові і отримувати інші результати. Таким чином студент навчається виконувати подібні завдання. Є можливість супроводжувати лекцію побудовою рисунків, графіків, зображенням поверхонь та інших графічних об'єктів.

Зручно використовувати такі лекції і для віддаленого навчання. Наприклад, лектор відкриває у себе на персональному комп'ютері зображення лекції і демонструє свій екран по Skype або іншим способом у навчальну аудиторію. Студенти в аудиторії бачать зображення на екрані і чують коментарі лектора.

4. Засвоєння нового матеріалу студентом

Студент може отримати попередньо всі файли з текстами лекцій. Прослухавши лекцію в університеті, він може прочитати її і запустити всі приклади, які демонструвалися в лекції. Для опанування нового матеріалу він повинен розв'язувати задачі. Розв'язування задач забезпечує засвоєння і закріплення нового матеріалу. Потрібно мати задачник, з якого можна буде брати умови задач на мові Mathpar. Для розв'язування задач він може скористатися сервісом MathPartner, як це робив лектор. Необхідні оператори він може просто копіювати з лекцій. Студент може уникнути виконання рутинних арифметичних дій і дій з громіздкими виразами завдяки вбудованим у MathPartner функціям.

Усі тексти розв'язування задач і їхні результати він може зберегти у вигляді текстового файлу. Для цього в лівому полі треба відкрити меню **«Файли»** і вибрати другий рядок **«Зберегти текст»**. Текст збережеться в папці завантаження. Треба тільки простежити, щоб браузер дозволив спливаючі вікна з сайту MathPartner.

Якщо ж щось залишається нерозв'язаним, то такий файл можна надіслати поштою викладачеві і попросити про допомогу та роз'яснення. Викладач завантажує цей файл і знаходить помилки студента, використовуючи MathPartner. Викладач звільняється від рутинних дій під час перевірки робіт студентів.

Звісно, в такому ж режимі можна проводити і контрольні роботи. Закінчивши виконання, студенти пересилають текстові файли з розв'язками контрольної роботи викладачеві.

5. Елементи синтаксису мови

В інших розділах ми описуємо основні конструкції мови Mathpar. Для більш детального ознайомлення з ними можна скористатися літературою [2; 3], сторінками «Допомоги» на сайті або завантажити «Керівництво з мови».

Процедури і функції

Mathpar є процедурною мовою програмування. Використовуються процедури і функції. Процедура починається зі слова *\procedure*, має ім'я і може мати аргументи, а оператор виходу з процедури може мати значення, що повертається: *\return(objectName)*, тоді така процедура є функцією. Синтаксис процедур і функцій такий:

\procedure proc1 (arg1, arg2, ..) {оператор 1; оператор 2; ..}

Оператори управління

Операторами управління є:

if () {} else {} – оператор розгалуження;

while () {} – оператор циклу з передумовою;

for (; ;) {} – оператор циклу з лічильником.

Є спеціальні оператори: оператори виведення значення виразів, оператори виведення графіків і оператори налаштування оточення.

Простір

У системі визначається простір, у якому відбуваються обчислення.

За замовчуванням визначено простір поліномів від чотирьох змінних над множиною дійсних чисел: *R64 [x, y, z, t]*, де $x<y<z<t$.

Для зміни простору треба виконати команду визначення нового простору. Наприклад, *SPACE = Q [x]* або *SPACE = Z [p, q]* і т. д.

Числові множини

Користувач може вибирати такі числові множини: *Z* – множина цілих чисел, *Zp* – скінченне поле *Z /pZ*, характеристика *p* задається користувачем у константі *MOD*; *Zp32* – скінченне поле *Z /pZ*, характеристика *p* задається константою *MOD32*. Різниця між *Zp* та *Zp32* полягає в тому, що характеристика в другому випадку повинна мати в записі не більше ніж 32 біти; *Z64* – підмножина цілих 64-бітних чисел; *Q* – множина раціональних чисел; *R* – множина дійсних чисел, у яких число цифр у мантисі задається користувачем (постійна ACCURACY); *R64* – множина дійсних чисел зі стандартною 52-розрядною мантисою і окремим 11-розрядним полем для зберігання порядку; *R128* – множина дійсних чисел, у яких 52 розряди в мантисі та 64 розряди використовуються для зберігання порядку.

Константи Mathpar

Можна встановлювати або змінювати значення констант, їхні значення за замовчуванням вказано в дужках. MOD32 (268435399) – характеристика поля *Zp32*. MOD (268435399) – характеристика поля *Zp*, вона може бути довільним простим числом. RADIAN (1) – визначає міру для кутів: радіани (1) або градуси (0). TIMEOUT (15) – максимальний час у секундах, який відводиться для обчислень. STEPBYSTEP (0) – режим виведення проміжних результатів. EXPAND (1) – режим розкриття всіх дужок у вхідному виразі. SUBSTITUTION (1) – режим заміни в кожному виразі всіх відомих змінних їхніми значеннями. FLOATPOS (2) – число десяткових знаків після коми, які виводяться на друк. MachineEpsilonR64 (36) – машинне епсилон у двійковому записі для чисел типу *R64*. ACCURACY (25) – число точних десяткових позицій після коми для чисел типу *R* і *C* при виконанні операцій множення, ділення і обчислення значень функцій. MachineEpsilonR (20) – машинне епсилон для десяткового запису чисел типу *R*. Повинно виконуватися: ACCURACY> MachineEpsilonR.

Основні класи функцій

Визначено такі символи: \i – уявна одиниця *i*, \e – число *e*, \pi – число π, \infty – знак нескінченності ∞, \emptyset – порожня множина ∅.

Визначено такі елементарні функції: \exp (), \lg(), \log_a(b), \sgrt(), \cubrt(), \rootOf(a, n), \sin(), \cos(), \tg() \ctg(), \arcsin(), \arccos(), \arctg(), \arcctg(), \sh(), \ch(), \th(), \cth(), \arcsh(), \arcch(), \arctgh(), \arcctgh(), \sc(), \csc(), \arcsec(), \arccsc().

Операції над числами: \max(a, b), \min(a, b), \sign(a), \abs(a), \floor(a), \ceil(a), \round(a).

Операції над цілими числами: \mod(a, m), \modInverse(a), a!, \divRem(a, b), \div(a, b), \rem(a, m).

Операції з дробом і раціональними функціями: \num (fr), \denom (fr), \cancel (fr), \properForm (fr), \quotientAndRemainder (fr), \quotient(fr),\remainder(fr).

Операції над многочленами *(f, g, f1, f2, .., fc)* багатьох змінних *(x, x0, y0, ..)*: \expand(f), \factor(f), \GCD(f, g), \LCM(f, g), \value(f, [x0, y0, ..]), \groebner(f1, f2, .., fc), \reduceByGB(g, [f1, f2, .., fc]), \solve(f(x)), \solve(f(x) = 0), \solveNAE (f1, f2, .., fc), \quotientAndRemainder(f, g), \quotient (f, g), \remainder(f, g). Оцінки складності алгоритмів над поліномами можна знайти в роботі [4].

Інші функції

\value (f, [x 0, y 0]) – підставити у функцію *f* значення змінних *x = x0, y = y0* та обчислити.

\value (g, [f0, f1]) – зробити заміну змінних у функції *g*: *x → f0, y → f1*.

\int (f)d x – інтеграл від *f* по *x* (первісна без довільної сталої),

\D (f, x) – похідна функції *f* за змінною *x*,

\D_{x^n} (f) – похідна порядку *n* функції *f* за змінною *x*,

\D_{x^n y^m} (f) – змішана похідна функції *f* за змінною *x* порядку *n*, за змінною *y* порядку *m*,

\lim_{x \to a} (f) – ліміт функції *f*, коли *x* прямує до *a*,

\binom (n, k) – число поєднань з *n* по *k*.

Матриці та вектори

Нульова матриця розміру *n* × *m* – *\O_{n, m}*,

матриця з одиницями на головній діагоналі – *\I_{n, m}*,

нульовий вектор, який має *n* компонент – *\O_{n}*.

Функції над матрицями:
\kernel (A), \transpose (A), \charPolynom (A), \transpose (A) або A^{T}, \conjugate(A) або A^{\ast}, \toEchelonForm(A), \det(A), \inverse() або A^{- 1}, \genInverse() або A^{+}, \adjoint () або A^{\star}, \closure() або A^{\times} = I + A + A ^ 2 + A ^ 3 + ...

Факторизація матриць над комутативною областю

\LDU (A) – трикутна LDU-факторизація матриці *A*. Результатом є п'ять матриць [*L, D, U, P, Q*]. Добуток *LDU* – це *A*. Тут *L* – нижня трикутна матриця, *U* – верхня трикутна матриця, *D* – матриця перестановок, помножена на діагональну матрицю, *P* і *Q* – матриці перестановок. При цьому *P^T LP* і *Q^T UQ* – це нижня і верхня трикутні матриці, *P^T DQ* – це діагональна матриця, у якої всі нульові діагональні елементи зібрані в нижніх рядках.

\BruhatDecomposition (A) – розкладання Брюа матриці. Результатом є три матриці [*V, w, U*], *V* і *U* – верхні трикутні матриці, *w* – матриця перестановок, помножена на діагональну матрицю, при цьому добуток *VwU* – це *A*. Тут використовувалися матричні алгоритми з робіт [5–8].

Перетворення Лапласа і розв'язання систем лінійних диференціальних рівнянь з постійними коефіцієнтами

\LaplaceTransform () – пряме перетворення Лапласа,

\InverseLaplaceTransform () – обернене перетворення Лапласа,

\SolveLDE(S, J) – розв'язок системи *S* лінійних диференціальних рівнянь з постійними коефіцієнтами при початкових умовах *J*. Ось приклади задання *S* і *J*:

S = \systLDE(\d (x, t) + x – 4y = t^2 \exp(2t)), 7 \d(y,t), – 2x – 2y = 3 \exp (3t) + t);

J = \initCond(\d (x, t, 0, 0) = 2, \d(y, t, 0, 0) = 0).

Тут використовуються позначення:

\D(f, t) – позначення для похідної функції *f* за змінною *t*.

\D(f, t, k) – позначення для *k*-тої похідної функції *f* за змінною *t*.

\D(f, t, k, t0) – позначення для *k*-тої похідної функції *f* за змінною *t* в точці *t0*.

Якщо *k = 0*, то це позначення для функції *f* у точці *t0*.

Якщо система диференціальних рівнянь дана в матричному вигляді, то можна використовувати оператор розв'язання з трьома аргументами: \solveLDE(A, B, C). Тут матриця *A* – це ліва частина системи диференціальних рівнянь після перетворення Лапласа, вектор *B* – права частина системи після перетворення Лапласа і *C* – це матриця початкових умов [9].

Функції теорії ймовірностей і математичної статистики

Функції неперервних випадкових величин

Для неперервної випадкової величини зі щільністю розподілу *f(x)*, яка задана на інтервалі *(a, b)*, визначено такі функції: *\mathExpectation(a, b, f(x))* – математичне сподівання, *\dispersion(a, b, f(x))* – дисперсія, *\meanSquareDeviation(a, b, f(x))* – середнє квадратичне відхилення.

Функції дискретних випадкових величин

Дискретна випадкова величина задається матрицею з двох рядків: у першому рядку записано значення випадкової величини, в другому – відповідні їм ймовірності. Сума всіх елементів другого рядка дорівнює одиниці. Наприклад, a = [[1,2,3,4,5], [0.4,0.1,0.1,0.2,0.2]]. Для дискретних випадкових величин визначено такі функції: *\mathExpectation()* – математичне сподівання, *\dispersion()* – дисперсія, *\meanSquareDeviation()* – середнє квадратичне відхилення, *\addQU(a, b)* – сума двох дискретних випадкових величин, *\multiplyQU(a, b)* – добуток двох дискретних випадкових величин, *\covariance(a, b)* – коефіцієнт коваріації двох дискретних випадкових величин, *\correlation(a, b)* – коефіцієнт кореляції двох дискретних випадкових величин, *\simplifyQU()* – спрощення запису дискретної випадкової величини, якщо в ній присутні значення, які повторюються.

Функції для вибірок

Вибірка задається як вектор, наприклад, S = [1,7,10,15]. Для вибірок визначені функції: *\sampleMean(S)* – вибіркове середнє, *\sampleDispersion(S)* – вибіркова дисперсія, *\covarianceCoefficient(S1, S2)* – коефіцієнт коваріації для двох вибірок, *\correlationCoefficient (S1, S2)* – коефіцієнт кореляції для двох вибірок.

Створення випадкових об'єктів: чисел, поліномів, матриць

Функція часу

\RandomNumber (k) – випадкове число, що містить *k* біт. *\RandomPolynom (nx, ny, nz, denP, k)* – випадковий поліном, у якого коефіцієнти – це числа, що містять *k* біт, степені за змінними *x, y, z* не перевищують *nx, ny, nz*, відповідно, і щільність полінома дорівнює *denP* відсотків. Щільність полінома – це відношення числа ненульових коефіцієнтів до максимально можливого числа коефіцієнтів.

\RandomMatrix (m, n, denM, k) – випадкова числова матриця $m \times n$ зі щільністю *denM* (відсотків), елементи якої – це *k*-бітні числа. Щільність матриці – це відношення числа ненульових елементів до загальної кількості елементів.

\RandomMatrix (m, n, denM, nx, ny, nz, denP, k) – випадкова поліноміальна матриця зі щільністю *denM*, елементи якої – випадкові поліноми, які створюються оператором *\randomPolynom(nx, ny, nz, denP, k)*.

\time() – час у мілісекундах. Щоб виміряти час деякого процесу, потрібно знайти різницю значень цієї функції, які отримані в кінці та на початку цього процесу.

Булева алгебра

Можна користуватися логічними операторами *\&, \lor, \neg* для позначення кон'юнкції, диз'юнкції і заперечення на множині булевих змінних, які приймають значення 1, 0. Ці ж оператори можна застосовувати і для матриць над булевими змінними.

Операції порівняння *\ge, ==, <, \ne, \le,* < позначають, відповідно, операції «більше або дорівнює», «більше», «дорівнює», «не дорівнює», «менше або дорівнює», «менше». Вони повертають значення *1 = true* і *0 = false*.

Алгебра множин на дійсних числах і на цілих числах

На множині дійсних чисел можна застосовувати бінарні операції: *\cup, \cap, \triangle, \setminus* та унарну операцію *(')*. Вони, відповідно, позначають об'єднання, перетин, симетричну різницю, віднімання множин та доповнення до всієї множини. Відкриті та замкнені компактні підмножини з межами *a* і *b* позначаються так: *\(a, b \)* і *\[a, b \]*. Можна застосовувати і напіввідкриті підмножини, за допомогою круглих та квадратних дужок.

Аналогічні операції можна виконувати і на множині цілих чисел.

6. Обчислення в тропічній математиці

Крім класичних числових алгебр з операціями +, -, * та операцією «ділити» для полів, можна робити обчислення в ідемпотентних алгебрах. У таких алгебрах перша операція ідемпотентна. У назві тропічної алгебри відразу вказуються операції алгебри. При виборі такої алгебри відбувається перепризначення символів + та * і порядку на числовій множині. Використовується порядок, узгоджений зі складанням: $x \leq y \Leftrightarrow x \oplus y = y$.

Нейтральний елемент по додаванню 0 є найменшим елементом по порядку, але при цьому він може відрізнятися від числового нуля [10].

Можна задавати такі ідемпотентні напівполя:

1) ZMaxPlus, ZMinPlus – на множині цілих чисел *Z*,

2) RMaxPlus, RMinPlus, RMaxMult, RMinMult – на множині дійсних чисел *R*,

3) R64MaxPlus, R64MinPlus, R64MaxMult, R64MinMult – на множині дійсних чисел *R64*.

Можна задавати такі ідемпотентні напівкільця:

1) ZMaxMin, ZMinMax, ZMaxMult, ZMinMult – на множині цілих чисел *Z*,

2) RMaxMin, RMinMax – на множині дійсних чисел *R*,

3) R64MaxMin, R64MinMax – на множині дійсних чисел *R64.*

Всього 18 різних типів алгебр. Ось приклади операторів, які встановлюють тропічне оточення: SPACE = ZMaxPlus[x, y, z]; SPACE = RMaxMin[u, v].

Приклад задання в напівкільці ZMaxPlus:

*a = 2; b = 9; c = a + b; d = a * b; \Print(c, d)*

У результаті отримаємо *c = 9, d = 11*. Відбулося перепризначення операцій: знаком плюс позначається бінарна функція максимум, а знаком помножити позначається операція додавання.

Крім бінарних операцій додавання і множення, використовується унарна операція замикання. Позначається вона так: *\closure(a)*, де *a* – це число або матриця. В результаті обчислюється вираз $1 + a^1 + a^2 + a^3 + \ldots$

Оператори в тропічних алгебрах

У тропічних алгебрах є такі оператори:

\solveLAETropic(A, b) – частковий розв'язок системи рівнянь $Ax = b$;

\BellmanEquation(A) – розв'язок однорідного рівняння Беллмана $Ax = x$;

\BellmanEquation(A, b) – розв'язок неоднорідного рівняння Беллмана $Ax + b = x$;

\BellmanInequality(A) – розв'язок нерівності $Ax \leq x$;

\BellmanInequality(A, b) – розв'язок нерівності $Ax \oplus b \leq x$.

Якщо для графа задана матриця *A* відстаней між суміжними вершинами, то для цієї матриці в алгебрах Min-Plus визначені ще два оператори:

\searchLeastDistances(A) – знаходить найкоротші відстані між усіма вершинами,

\findTheShortestPath(A, i, j)) – знаходить найкоротший шлях між вершинами *i* та *j*.

7. Додатковий сервіс

Побудова 2D графіків функцій

Графічне оточення задається командою *\set2D()*. Повний формат для команди *\set2D()* передбачає 3 групи параметрів, кожна з яких пишеться у квадратних дужках: \set2D([x0, x1, y0, y1], ['xTitle', 'yTitle', 'title'], [0,1,12,3,5]). Перша квадратна дужка визначає межі графіка. Друга квадратна дужка – це підписи до осей координат і підпис до всього малюнка. Третя з квадратних дужок містить 5 чисел: 1) 1 – означає: встановити режим чорно-білий (0 – кольоровий), 2) 1 – означає: встановити рівний масштаб по обох осях (0 – золотий перетин), 3) це розмір шрифту для підписів, 4) це товщина ліній графіків, 5) це товщина координатних осей. Будь-яка з пар квадратних дужок, крім першої, може не задаватися. Параметрами зображуваної лінії можуть бути: dash (пунктир), arrow (стрілки) або dashAndArrow. Ці параметри можуть стояти в кінці списку параметрів функцій \plot, \tablePlot і \paramPlot. Наприклад, \plot(cos (x), dash).

Три способи задання графіків: *\plo(f)*, де *f = f (x)* – функція, задана явно, *\paramPlot([f, g], [xMin, xMax]),* де *f = X (x), g = Y (x)* – функції, задані параметрично, *\tablePlot(T)*, де *T = [[x1, .., xn], [f1, .., fn], .., [g1, .., gn]]* – це одна або кілька таблично заданих функцій, записаних у вигляді матриці: перший рядок – значення абсцис точок, а другий і наступні – значення функцій у цих точках. Кожна функція записується в один рядок. Для підписів на графіках застосовується оператор *\textPlot()*, а для виведення ізольованих точок – *\pointPlot()*.

Можна зобразити кілька графіків на одному малюнку. Для цього потрібно заздалегідь визначити ці графіки, наприклад: *P1 = \plot(x^2)*; *P2 = \plot(x^3 + 1)*. Оператор \showPlots ([P1, P2]) – дає зображення декількох графіків на одному малюнку. Параметри, які можуть стояти в кінці всіх параметрів showPlots: 'noAxes' – виведення графіка без осей і 'lattice' – виведення графіка з сіткою.

Побудова 3D графіків функцій

Для функцій $z = f(x, y)$, які задані явно, можна побудувати графік так: \plot3d(f, [xMin, xMax, yMin, yMax]) і \explicitPlot3d (f, xMin, xMax, yMin, yMax, zMin, zMax). Для функцій, які задані неявно $f(x, y, z) = 0$, можна побудувати графік так: \implicitPlot3d(f, xMin, xMax, yMin, yMax, zMin, zMax).

Можна додатково вказувати координати джерела світла (**lightX, lightY, lightZ**), колір (**color**) і число точок на сітці (**gridSize**). За замовчуванням приймається на сітці 50 точок на кожному ребрі паралелепіпеда. Колір у форматі RGB (червоний, зелений, блакитний) задається числом $color = R * 256 * 256 + G * 256 + B$, де кожна буква позначає невід'ємне ціле число, яке не перевищує 255. Повний набір параметрів для цієї функції такий: \implicitPlot3d (f, xMin, xMax, yMin, yMax, zMin, zMax, lightX, lightY, lightZ, color, gridSize).

Виведення і збереження результатів

Поле, в якому з'являються результати обчислень, розташоване в робочому зошиті у вікні браузера відразу під вікном введення. Сервіс робочого зошита забезпечує додавання або видалення додаткових вікон. У кожному такому окремому вікні можна виконувати команди і отримувати результати обчислень незалежно від інших. При цьому всі раніше створені об'єкти зберігаються в пам'яті і можуть бути використані в будь-якому вікні. Є можливість зберегти в одному текстовому файлі всі поля вводу та виводу, які є в робочому зошиті. При завантаженні такого файлу на сервер відбудеться відновлення всіх вікон відразу. Користувач має можливість зберегти вікна в мові Mathpar, в мові LaTeX або в pdf-файлі.

Крім цього, є можливість зберегти один математичний об'єкт у текстовому файлі користувача. При виконанні оператора *\toFile(f, fileName)* відбувається запис об'єкта *f* в мові Mathpar і збереження цього файлу на комп'ютері користувача.

При цьому сам об'єкт на екран не виводиться. Таким чином досягається збереження великих об'єктів, запис яких у мові Mathpar перевищує розміри екрана.

Сервіс для обчислень на кластері

Налаштування параметрів завдання для обчислення на кластері забезпечує зв'язок з обчислювальним кластером і надає можливість для виконання паралельних обчислень на кластері для окремих операторів.

Для обчислень на кластері потрібно задати постійні налаштування завдання на кластері:

TOTALNODES – загальна кількість вузлів кластера, які виділяються для обчислень,

PROCPERNODE – кількість MPI-процесів, що запускаються на одному вузлі,

CLUSTERTIME – максимальний час (у хвилинах) виконання програми, після закінчення якого програма примусово завершиться.

MAXCLUSTERMEMORY – обсяг пам'яті, що виділяється для JVM для одного MPI-процесу (опція -Xmx). Оскільки загальна оперативна пам'ять на вузлі ділиться порівну між усіма його ядрами, то за рахунок зміни кількості ядер можна змінювати обсяг пам'яті, який припадає на одне ядро.

На сьогодні підключені такі паралельні оператори: для класичних просторів – це множення матриць і обчислення оберненої матриці (приєднаної матриці та визначника) \adjointDetPar(A); для тропічних просторів R64MaxPlus[x]: \BellmanEquationPar(A), \BellmanEquationPar(A, b), \BellmanInequalityPar(A, b).

Символи, які застосовуються в математичних текстах

Наведемо математичні символи, які можна використовувати в тексті на мові Mathpar.

∂ =\partial; ∇ =\nabla; $\hbar$ =\hbar; $\to$ =\to; $\perp$ =\perp; $\|$=\parallel; $\angle$ =\angle; $\smile$ =\smile; $\equiv$ =\equiv\; $\square$ =\square; $\blacksquare$ =\blacksquare; $\approx$ =\approx; $\sim$ =\sim; $\in$ =\in; $\notin$ =\notin; $\ni$ =\owns; $\subset$ =\subset; $\subseteq$ =\subseteq; $\supset$ =\supset; $\supseteq$ =\supseteq; $\exists$ =\exists; $\forall$ =\forall; $\vee$ =\vee; $\wedge$ =\wedge; $\oplus$ =\oplus; $\otimes$ =\otimes; \degreeC – верхній кружечок, \circ – центральний кружечок. Надрядкові символи: a^ =\hat; $\bar{a}$ =\bar; a~ =\tilde; $\vec{a}$ =\vec; a =\dot; a =\ddot. Надрядкові групові символи: \widetilde{}; \widehat{}; \overline{}; \overrightarrow{}.

Крім того, *\underbrace{}* і *\overbrace{}* – це позначення для верхньої фігурної дужки і нижньої фігурної дужки, *\system(, ,)* і *\systemOR(, ,)* – це позначення для системи рівнянь з фігурною дужкою і з квадратною дужкою, в яких окремі рівняння розділяються комами, *\frac{a}{b}* – це запис дробу, де *a* – чисельник і *b* – знаменник.

Всі ці символи можна використовувати в текстовій частині, оточуючи знаками $.

Ці та інші підказки для символів, наприклад, грецькі літери, можна знайти в лівій панелі на сайті mathpartner.ukma.edu.ua в розділі Symbols.

Висновки

Застосування хмарної математики MathPartner може привести до істотної інтенсифікації освіти в тих сферах, де використовується математика. Крім того, хмарна математика є універсальним інструментом застосування математичних знань і може сприяти інтенсивному розвитку прикладної математики.

Список літератури

1. Malaschonok G. I. Way to Parallel Symbolic Computations [Електронний ресурс] / G. I. Malaschonok // Облачные вычисления. Образование. Исследования. Разработка : [материалы конференции]. – Москва, 2011. – Режим доступу: www.unicluster.ru/conf/2011/docs/TSU.MalaschonokG.I.pdf. – Назва з екрана.
2. Малашонок Г. И. О проекте параллельной компьютерной алгебры / Г. И. Малашонок // Вестник Тамбовского университета. Серия: Естественные и технические науки. – Тамбов, 2009. – Т. 14, вып. 4. – С. 744–748.
3. Малашонок Г. И. Система компьютерной алгебры MathPartner / Г. И. Малашонок // Программирование. – 2017. – № 2. – С. 63–71.
4. Малашонок Г. И. О выборе алгоритма умножения для полиномов и полиномиальных матриц / Г. И. Малашонок, Ю. Д. Валеев, А. О. Лапаев // Записки научных семинаров ПОМИ. – 2009. – Т. 372. – С. 50–82.
5. Malaschonok G. I. Effective matrix methods in commutative domains / G. I. Malaschonok // Krob D. Formal Power Series and Algebraic Combinatorics / D. Krob, A. A. Mikhalev, A. V. Mikhalev (eds.). – Berlin : Springer, 2000. – P. 506–517.
6. Akritas A. G. Computations in Modules over Commutative Domains / A. G. Akritas, G. I. Malaschonok // Computer Algebra in Scientific Computing. – Berlin : Springer, 2007. – P. 11–23.
7. Malaschonok G. I. Generalized Bruhat decomposition in commutative domains / G. I. Malaschonok // International Workshop on Computer Algebra in Scientific Computing, LNCS 8136. – Berlin, Heidelberg : Springer, 2013. – P. 231–242.
8. Malaschonok G. Triangular Decomposition of Matrices in a Domain / G. Malaschonok, A. Scherbinin // Computer Algebra in Scientific Computing. LNCS 9301. – Springer, 2015. – P. 290–304.
9. Malashonok N. A. Symbolic-Numerical Solution of Systems of Linear Ordinary Differential Equations with Required Accuracy / N. A. Malashonok, M. A. Rybakov // Programming and Computer Software. – 2013. – Vol. 39, no. 3. – P. 150–157.
10. Киреев С. А. Тропические вычисления в веб-сервисе MathPartner / С. А. Киреев, Г. И. Малашонок // Вестник Тамбовского университета. Серия: Естественные и технические науки. – Тамбов, 2014. – Т. 19, вып. 2. – С. 539–550.

G. Malaschonok

CLOUD MATHEMATICS MATHPARTNER IN KYIV-MOHYLA ACADEMY

Recently, cloud services have been replacing application packages. The same process occurs in systems of symbolic computations. In the last decade, such SaaS systems of symbolic computations as MathPartner, Sage, SMathStudio, Wolfram Alpha and Wolfram Cloud have appeared.

The number of users of systems of symbolic computations in the total number of those who study and apply mathematics has so far been insignificant. This situation is changing rapidly nowadays due to the emergence and wide application of SaaS systems. Cloud mathematical services are expected to will lead to a change in the status of mathematical knowledge in the modern society and to a drastic change in the entire education system. Mathematical knowledge will be easy to apply, with avoiding routine calculations.

One of the first cloud systems was MathPartner, which appeared in 2011. Today this service is installed in the Kyiv-Mohyla Academy and is available at http://mathpartner.ukma.edu.ua.

Mathpar is the language of the MathPartner service. The text library in Mathpar is located in the GitHub repository. This library is freely available at http://github.com/mathpar.

One can register on the GitHub web service and after that post one's texts and receive texts posted by other participants. One can download the entire library archive as a zip file: https://github.com/mathpar/mathpar/archive/master.zip.

This is the root of the archive https://github.com/mathpar/mathpar/tree/master/LIBRARY/mathpar. For example, if a user selects the path EUROPE/Ukraine-ua/UNIVERSITY/MATH/ Computer _Algebra/Malaschonok/Computer_Algebra_Chap_01, then all paragraphs of the first chapter of the textbook on computer algebra will be available. One can open and read any of them. One can copy all the text. The service supports receiving and editing code via Git, SVN, and Mercurial.

This article describes MathPartner and the new opportunities that it provides to a wide range of users: from professionals of mathematicians to junior schoolchildren, from theoretical physicists to mathematics and physics teachers.

Keywords: cloud mathematics, computer algebra system, MathPartner, symbolic computations.

Матеріал надійшов 18.10.2017