

## **ТЕМА: Розв'язування задачі Коші для звичайних диференціальних рівнянь першого порядку засобами пакету Mathcad**

Багато фізичних систем та явищ описуються рівняннями, в які входять не тільки невідома функція, а й деякі похідні від цієї функції. Як відомо, такі рівняння називаються диференціальними. У випадку коли шукана функція заданого диференціального рівняння залежить лише від однієї незалежної змінної (аргументу), то таке рівняння називається звичайним диференціальним рівнянням. Порядком звичайного диференціального рівняння називається найвищий порядок похідної, що входить у це рівняння. У найпростішому випадку розглядаються диференціальні рівняння першого порядку, тобто рівняння, які не містять похідних невідомої функції від незалежної змінної вище першого порядку. В загальному випадку такі рівняння можна записати у вигляді

$$y' = f(x, y) \quad (1)$$

де  $y = y(x)$ - невідома функція від незалежної змінної  $x$ ,  $f(x, y)$ - деяка задана функція. Розв'язком заданого диференціального рівняння називається деяка функція, яка задовільняє це рівняння у кожній точці області визначення рівняння. Як відомо з курсу диференціальних рівнянь [1], будь-яке звичайне диференціальне рівняння має нескінченну множину розв'язків, або, як прийнято казати, розв'язок звичайного диференціального рівняння визначається з точністю до константи. Така ситуація є не прийнятною з практичної точки зору, оскільки в конкретній фізичній задачі дослідника цікавить лише один розв'язок. Для виокремлення цього розв'язку з нескінченної множини розв'язків на нього накладаються деякі додаткові умови. Прийнято розрізняти два типи додаткових умов: початкові умови та граничні умови. У будь-якому випадку ці умови дозволяють отримати єдиний розв'язок, який повинен задовільняти як саме диференціальне рівняння, так і ці додаткові умови.

Початкові умови задаються у випадку коли незалежна змінна розглядається як час, а процес, який описується заданим рівнянням носить нестационарний характер. Тоді для адекватного математичного опису такого процесу не достатньо мати саме рівняння, потрібно задати стан процесу в деякий початковий момент часу. Оскільки стан процесу в довільний момент часу характеризується невідомою функцією  $y$ , то задання початкового стану даного процесу, тобто початкової умови, полягає в заданні значення функції  $y$  в початковий момент часу  $x_0$ .

Сама задача знаходження розв'язку заданого звичайного диференціального рівняння, який задовільняє заданій початковій умові називається задачею Коші. Для прикладу, сформулюємо строго математично задачу Коші для звичайного диференціального рівняння першого порядку: знайти функцію  $y = y(x)$ , яка задовільняє рівняння (1) та початковій умові

$$y(0) = y_0. \quad (2)$$

Якщо диференціальне рівняння має порядок вище першого, то необхідно задавати початкове значення не тільки самій функції, а й її похідним. Загальне правило можна сформулювати наступним чином: якщо у звичайне диференціальне рівняння входить похідна шуканої функції  $n$ -го порядку, то початкових умов має бути  $n$  і задаються вони на саму функцію та її похідні до  $n - 1$  порядку включно.

Граничні умови задаються тоді, коли незалежна змінна трактується як просторова координата. Очевидно, що у випадку звичайного диференціального рівняння це означає, що розглядається одновимірна задача, в якій область визначення представляє собою відрізок, а границя цієї області визначення складається з двох точок. Тоді на кожному кінці відрізка потрібно задати граничну умову. Кількість граничних умов, які необхідно задати залежить від порядку диференціального рівняння. У загальному випадку, якщо порядок рівняння  $2n$ , то у кожній точці границі області визначення задачі потрібно задати  $n$  граничних умов. Сама задача знаходження розв'язку заданого звичайного диференціального рівняння, який задовільняє заданим граничним умовам називається крайовою задачею.

У ранніх версіях системи MATHCAD для розв'язання зазначених задач доводилось самотійно складати програму, яка реалізовувала той чи інший чисельний або аналітичний метод. Такий підхід був далеким від досконалого, оскільки вимагав досконалих знань в області чисельних методів, і, звичайно, був не прийнятним для інженерів-фахівців своєї галузі. Починаючи з версії MATHCAD 5.0 PLUS, у системі введено можливість чисельного розв'язання як окремих звичайних диференціальних рівнянь, так і їх систем. Цю можливість важко переоцінити, оскільки розв'язання великої кількості науково-технічних задач базується на чисельних методах розв'язання диференціальних рівнянь.

Для розв'язання звичайних диференціальних рівнянь першого порядку в системі MATHCAD найчастіше використовується функція *rkfixed*. Для ілюстрації синтаксису та правил використання даної функції розглянемо наступну задачу Коші:



вектор початкових умов функції  $g_{kfixed}$  повинен мати розмірність рівну порядку рівняння і складатися із початкових значень функції та її похідних до певного порядку.

Наступні два параметри задають початкове та кінцеве значення проміжку інтегрування, відповідно. Слід зауважити, що початкові умови, які задані у першому параметрі – це значення шуканого розв'язку в точці, що відповідає початковому значенню проміжку інтегрування.

Усі чисельні методи розв'язання диференціальних рівнянь базуються на так званій дискретизації вихідної задачі. Перший крок цього процесу полягає в тому, що неперервна область визначення задачі (у нашому випадку це відрізок  $[0;1]$ ) покривається множиною точок, які називаються вузлами. Сукупність цих вузлів часто називається сіткою. Тоді розв'язати вихідну задачу означає знайти значення розв'язку в вузлах сітки, а різні чисельні методи відрізняються один від одного лише способом отримання цих значень. Для звичайних диференціальних рівнянь проміжок інтегрування представляє собою простий відрізок, дискретизація якого полягає в розбитті на фіксовану кількість менших відрізків, які повинні взаємно не перетинатися і мати спільні точки. Задання цієї кількості частин, або те саме, що й кількості точок - вузлів сітки, в яких шукатимуться значення невідомого розв'язку – це і є суть значення наступного, четвертого, параметра функції  $g_{kfixed}$ . За допомогою цього параметру задається також розмірність матриці, яка буде повернена функцією  $g_{kfixed}$ .

Останній, п'ятий, параметр відіграє найважливішу роль. За його допомогою задається саме диференціальне рівняння, яке необхідно розв'язати. Цей параметр повинен бути обов'язково попередньо описаний як функція, яка повертає значення у вигляді вектору розмірності, що рівна порядку диференціального рівняння. Вся складність задання цього параметра полягає в тому, що необхідно уніфікувати спосіб представлення самого рівняння таким чином, щоб це представлення не залежало від порядку рівняння, значення коефіцієнтів, наявності вільного члену і т.д. Цього можна досягнути шляхом зведення вихідного диференціального рівняння  $n$ -ого порядку до системи з  $n$  рівнянь першого порядку відносно іншої, нової функції, кожне з яких розв'язане відносно першої похідної, тобто має вигляд (1). Тоді праві частини цих рівнянь і будуть елементами вектору, який має повертати функція, що задає цей останній параметр. У нашому прикладі диференціальне рівняння має перший порядок, тому вектор похідних вироджується в один елемент. Залишилося розв'язати саме рівняння відносно першої похідної, у результаті чого

воно набуде вигляду  $y' = -Ay$ . Права частина цього рівняння й утворюватиме тіло нашої функції  $D(x, y)$ . Слід зауважити, що першим аргументом функції визначення вектору похідних  $D(x, y)$  завжди має бути ім'я незалежної змінної навіть, якщо сама змінна не входить явно в рівняння (як у нашому випадку).

Розглянемо тепер яким чином правильно задати функцію  $D(x, y)$  у випадку звичайного диференціального рівняння другого порядку на прикладі такого рівняння

$$4y'' + 4y' + 9y = -27 \cos(3x) - 12 \sin(3x). \quad (5)$$

Введемо в розгляд дві функції  $y_0(x)$  та  $y_1(x)$  так, що

$$y_0(x) = y(x), \quad y_1(x) = y'_0(x).$$

Тоді рівняння (5) можна записати наступним чином

$$4y'_1 + 4y_1 + 9y_0 = -27 \cos(3x) - 12 \sin(3x). \quad (6)$$

Отже, тепер наше рівняння містить дві невідомі функції  $y_0(x)$  та  $y_1(x)$ , які, однак, чітко взаємопов'язані одна з одною. А це означає, що ми звели вихідне рівняння (5) другого порядку до наступної системи рівнянь першого порядку

$$\begin{cases} y_1(x) = y'_0(x) \\ 4y'_1(x) + 4y_1(x) + 9y_0(x) = -27 \cos(3x) - 12 \sin(3x) \end{cases} \quad (7)$$

Залишилося розв'язати кожне рівняння системи (7) відносно першої похідної, тобто привести до вигляду

$$\begin{cases} y'_0(x) = y_1(x) \\ y'_1(x) = -\frac{1}{4}(27 \cos(3x) + 12 \sin(3x) + 4y_1(x) + 9y_0(x)) \end{cases}, \quad (8)$$

і задати функцію визначення вектору похідних  $D(x, y)$  так, як показано на рис.2, який містить фрагмент документа MATHCAD для розв'язання рівняння (5) при наступних початкових умовах

$$y(0) = 5, \quad y'(0) = -2. \quad (9)$$

Як видно з рис.2, вектор початкових умов має розмірність 2, причому перший елемент відповідає початковому значенню функції, а другий – початковому значенню першої похідної від функції.

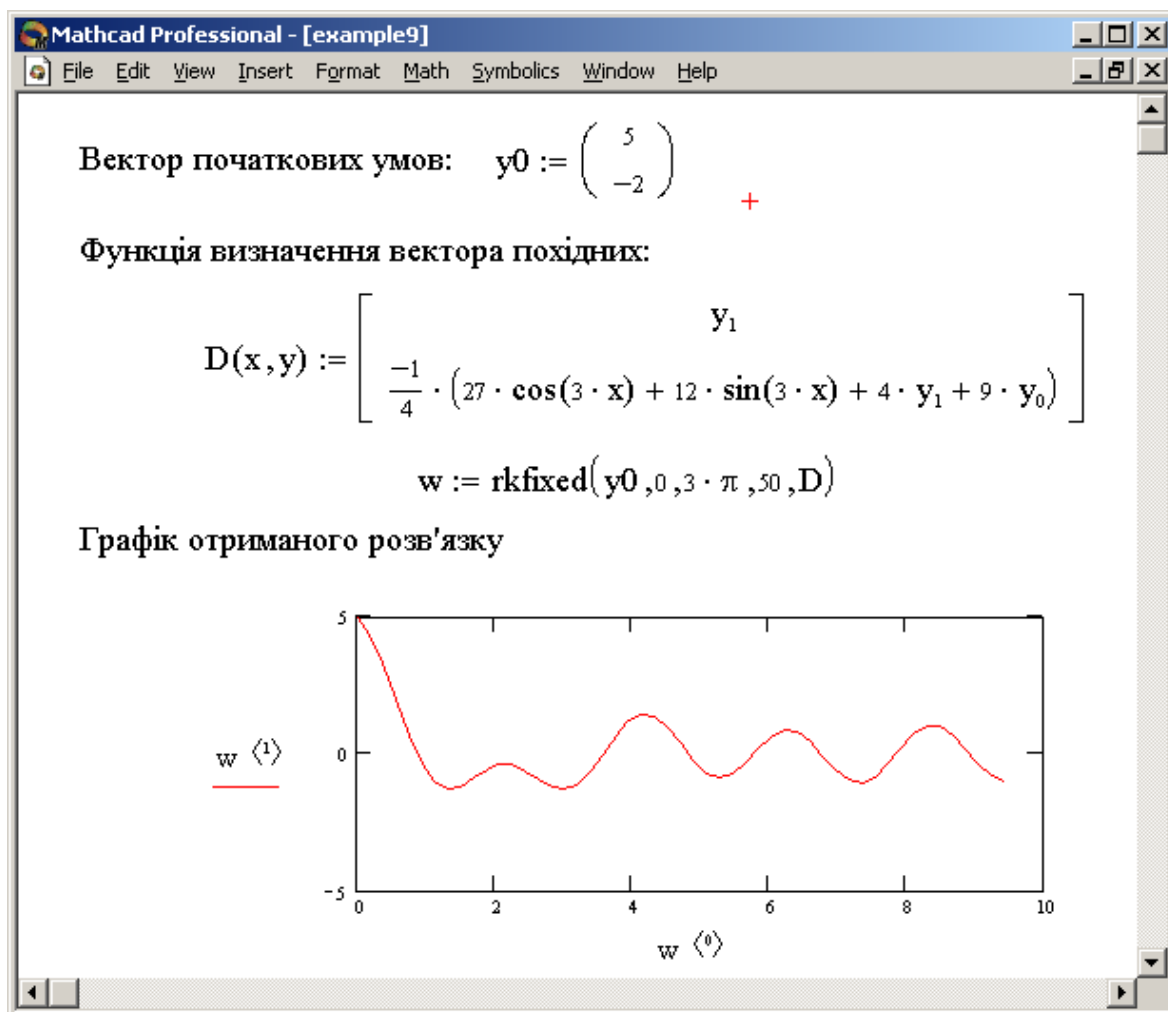


Рис.2. Приклад використання функції *rkfixed* для розв'язання звичайного диференціального рівняння другого порядку

Результатом виконання функції *rkfixed* є матриця такої структури: перший стовпець містить значення координат вузлів, у яких шукається розв'язок, другий стовпець – вузлові значення шуканої функції, третій і при потребі кожний наступний – вузлові значення похідних від шуканої функції, відповідно. Кількість рядків результуючої матриці визначається четвертим параметром функції *rkfixed* (див. вище опис цього параметра). Кількість стовпців матриці задається порядком диференціального рівняння, вона рівна значенню порядку плюс 1. Так, наприклад, для рівняння (5) з початковими умовами (9) та значеннями параметрів функції *rkfixed* як зображено на рис.2 результуюча матриця буде мати розмірність  $50 \times 3$  (слід пам'ятати, що в системі MATHCAD нумерація рядків та стовпців матриць починається з нуля). Такий спосіб представлення результату є зручним і дозволяє легко будувати графік отриманого чисельного розв'язку, як це зображено на рис.1-2 (за вісь абсцис відповідає перший стовпець  $w^{(0)}$ , а за вісь ординат – стовпець  $w^{(1)}$ ).

Розв'язання звичайних диференціальних рівнянь порядку  $n > 2$  здійснюється аналогічно. У цьому випадку вектор початкових умов буде мати розмірність рівну порядку рівняння і буде містити початкові значення шуканої функції та її похідних до  $n - 1$ -ого порядку включно. Вектор похідних тепер буде також мати розмірність  $n$  і буде складатися з похідних першого порядку від кожної нової функції системи з  $n$  диференціальних рівнянь першого порядку.

Слід зауважити, що функція `rkfixed` реалізує метод Рунге-Кутта з фіксованим кроком інтегрування (відрізок на якому розв'язується рівняння розбивається на рівновіддалені один від одного вузли з незмінним кроком розбиття  $h$ ). На практиці досить часто виникають ситуації, які вимагають використання нерівномірних(адаптивних) сіток, тобто сіток, в яких крок розбиття може змінюватися. Прикладами таких ситуацій можуть бути сингулярність розв'язку або наявність сильного локального градієнта розв'язку. В таких випадках нерівномірні сітки дозволяють отримати наближений розв'язок з вищою точністю у порівнянні з рівномірними сітками. МATHCAD містить функцію *rkadapt*, яка реалізує метод Рунге-Кутта із змінним кроком інтегрування, причому цей крок інтегрування змінюється автоматично в залежності від потреби.

Функція *rkadapt* має наступний синтаксис:

$$rkadapt(y,a,b,eps,D,k,s),$$

де значення перших трьох параметрів  $y, a, b$  повністю ідентичні першим трьом параметрам функції `rkfixed`, тобто це є вектор початкових умов, початкове та кінцеве значення проміжку інтегрування, відповідно,  $eps$  – точність обчислення наближеного розв'язку,  $D$  – функція, яка повертає вектор похідних (аналогічно як і для функції `rkfixed`),  $k$  – максимальна кількість точок інтегрування (оскільки крок змінюється, то наперед визначити точну кількість вузлів, в яких шукатиметься значення невідомого розв'язку неможливо; можна задати лише верхню межу кількості вузлів),  $s$  – мінімально допустиме значення кроку інтегрування (накладає обмеження на інтервал між сусідніми вузлами в результуючій матриці, хоча при потребі, для досягнення заданої точності, функція `rkadapt` може обчислювати крок, величина якого може бути меншою за вказане значення).

Результатом виконання функції `rkadapt` є матриця, структура якої повністю ідентична структурі результуючої матриці функції `rkadapt`. Рис.3 ілюструє приклад використання функції `rkadapt` для розв'язання такої задачі Коші

$$y'' + y'^2 = 0, \quad 1 \leq x \leq 15 \quad (10)$$

$$y(1) = 0, \quad y'(1) = 1$$

з точністю  $eps = 0.001$  (точний розв'язок задачі (10)-  $y(x) = \ln x$ ).

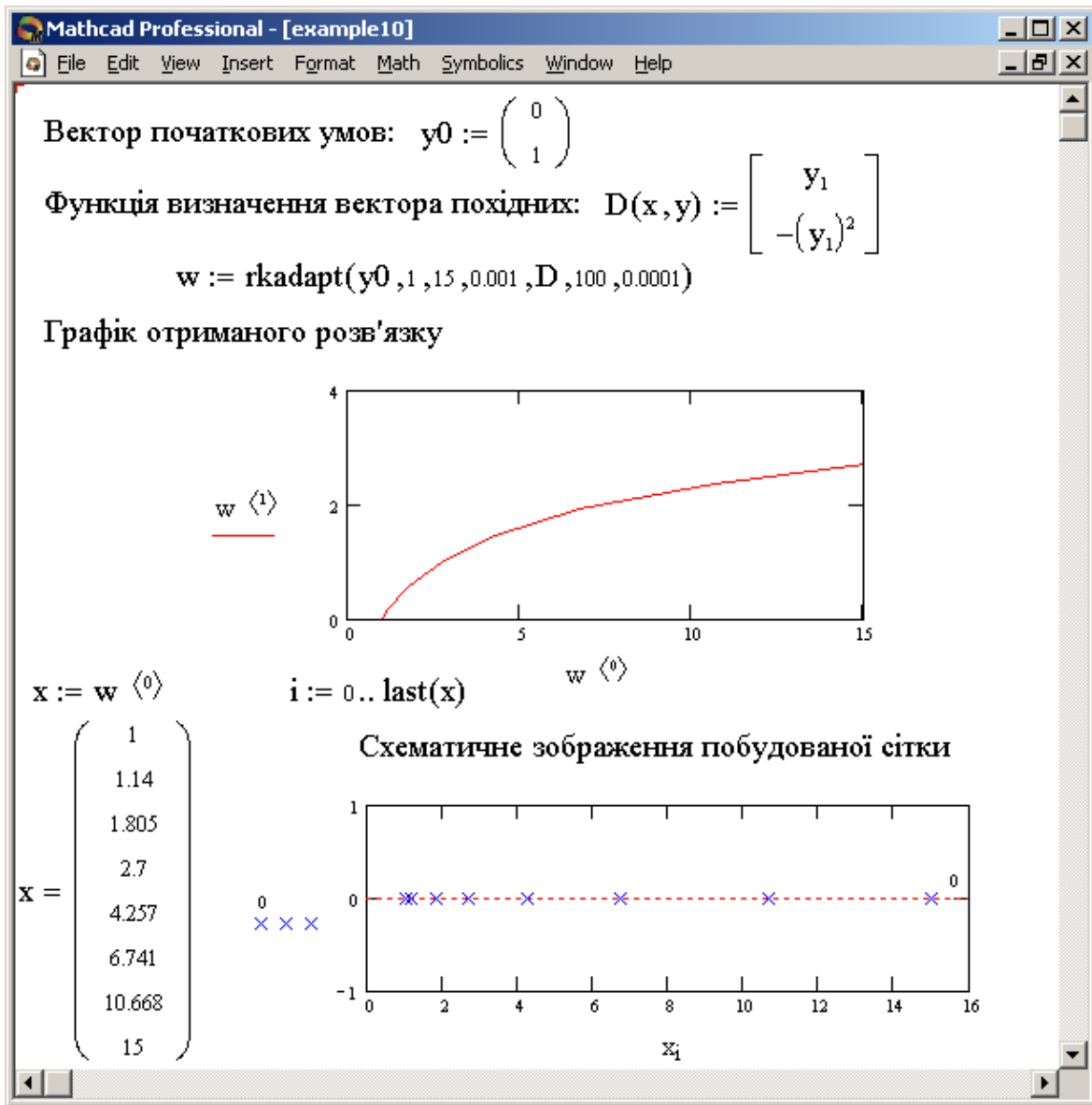


Рис.3. Приклад використання функції *rkadapt*

На цьому ж рисунку схематично зображено побудовану функцією *rkadapt* адаптивну сітку, з якої чітко видно, що поблизу лівого кінця проміжку інтегрування сітка значно густіша, ніж поблизу правого кінця (що цілком зрозуміло, адже функція  $\ln x$  значно швидше змінюється на початку своєї області визначення).

Починаючи з версії MATHCAD 2000 Professional для розв'язання звичайних диференціальних рівнянь була введена нова функція *odesolve*, яка значно спрощує процес отримання розв'язку. Це спрощення полягає в символному заданні як самого диференціального рівняння, так і по-



чаткових умов у спеціальному обчислювальному блоці, який починається з директиви *Given*. Функція *odesolve* має три параметри: перший параметр задає ім'я незалежної змінної, другий – кінцеве значення проміжку інтегрування, третій, необов'язковий, параметр визначає кількість точок інтегрування. Причому, якщо цей третій параметр задано, то розв'язання рівняння здійснюється з фіксованим кроком інтегрування, у протилежному випадку – з адаптивним кроком. У будь-якому випадку розв'язок заданого диференціального рівняння шукається у вигляді функції. Хоча аналітичний вираз для цієї функції у явному вигляді не виводиться, над нею можна виконувати різні математичні перетворення, наприклад, диференціювати, будувати графік і т.д. На рис.4 показано приклад застосування функції *odesolve*.

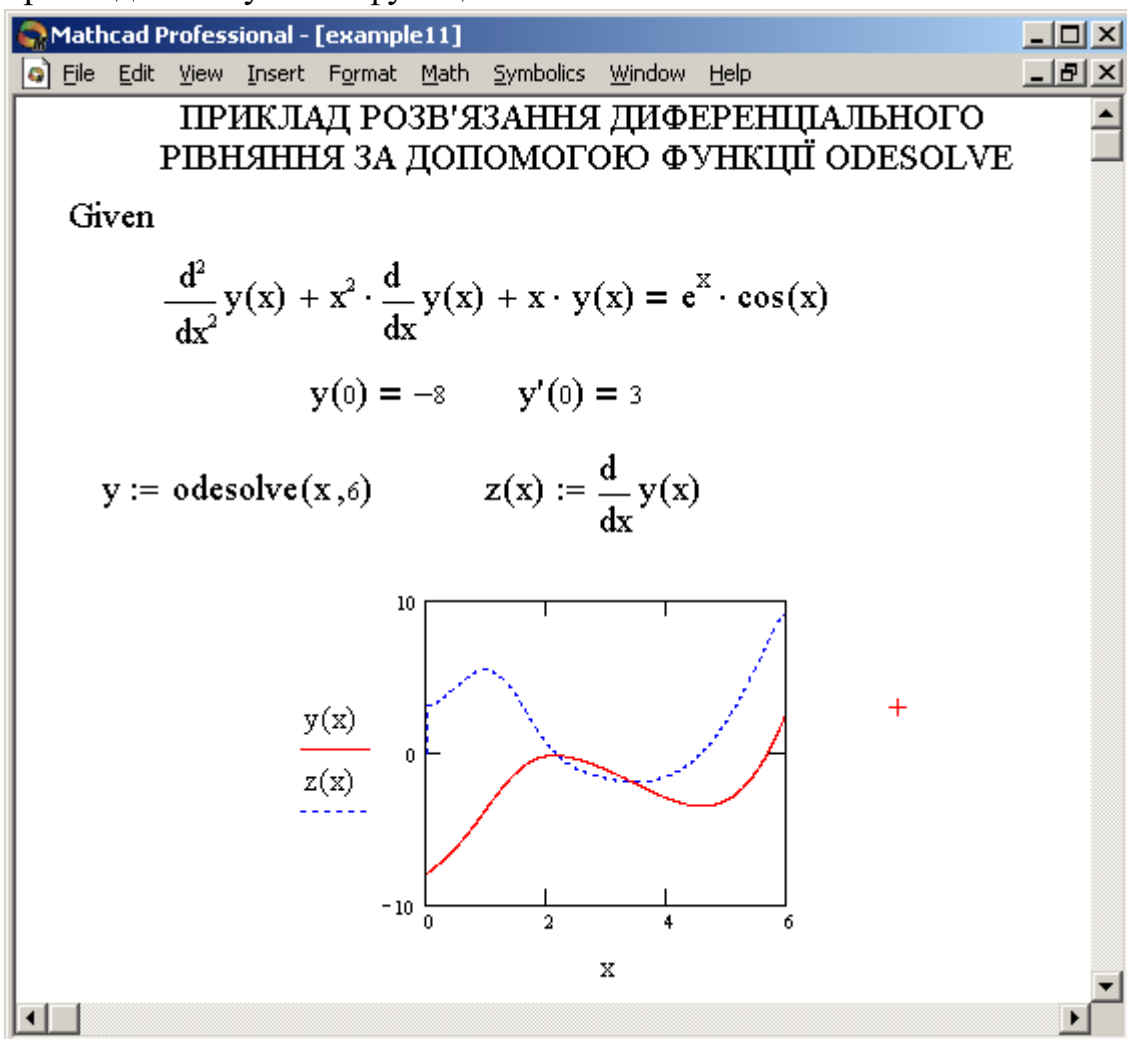


Рис.4. Приклад використання функції *odesolve*

Як видно з рис.4, при використанні функції *odesolve* диференціальне рівняння і початкові умови задані у звичному, зрозумілому вигляді, що значно полегшує розуміння та читабельність документа. Слід також

зауважити, що початкове значення проміжку інтегрування функція `odesolve` визначає автоматично з початкових умов.

Описаними вище функціями не вичерпується набір засобів розв'язання звичайних диференціальних рівнянь в системі MATHCAD. Зокрема, існує цілий ряд вбудованих функцій для розв'язання так званих жорстких систем диференціальних рівнянь, які відрізняються одна від одної різними методами розв'язання. Передбачено також декілька функцій для розв'язання двоточкових крайових задач для звичайних диференціальних рівнянь. Детальний опис всіх цих функцій з великою кількістю прикладів їх застосування можна знайти в центрі інформаційних ресурсів *Resource Center* системи MATHCAD, який викликається одноіменною командою меню *Help*.