Практичне заняття № 8.

Тема. Комп'ютерні процедури диференціювання та інтегрування функцій.

Мета роботи: навчитись проводити операції диференціювання та інтегрування за допомогою команд **MATLAB**; ознайомитись з процедурами комп'ютерного диференціювання та інтегрування функцій

Теоретичний мінімум

Формули для чисельного обчислення похідних важливі в розробці алгоритмів при обчисленні граничних значень в задачах, що зв'язані з звичайними диференціальними рівняннями в частинних похідних.

Нагадаємо, що похідною функції $y = f(x)$ по незалежному аргументу x в точці x_0 називається границя, до якої прямує відношення приріcту функції Δy к приросту аргументу Δx, коли приріст аргументу прямує до нуля. Операція знаходження похідної функції має назву *операції диференціювання*[51].

Основною задачею диференціального счислення є визначення для заданої функції $f(x)$ її похідної (або старших похідних) $f'(x) = \dfrac{dy}{dx}$ і диференціала $F'(x)dx = f(x)dx$.

Зворотна задача – визначення функції $F(x)$ по її формулам длявідомим похідної $f(x)$ або диференціалу $f(x)dx$. Це є основна задача інтегрального вирахування.

Первісною функції $y = f(x)$, яка визначена на відрізку $[a, b]$, називається функція $F(x)$, яка визначена на тому ж самому відрізку та задовільнює умові $F'(x) = f(x)$ або $dF(x) = f(x)dx$.

Процес знаходження первісної функції називається інтегруванням.

Завдання 1. Приклади диференціювання у системі **MATLAB**.

Обчислення інтегралів та похідних у багатьох випадках досить рутинна операція при використанні пакетів аналітичних обчислень (маються на увазі пакети символьної математики). Разом з тим часто задача містить числові дані, тому для роботи необхідні ефективні обчислювальні процедури.

Як уже відомо, у системі **MATLAB** є підпакет, який призначений для аналітичних обчислень – це **Symbolic Math Toolbox** [44]. У цьому пакеті існує команда **diff**, яка обчислює скінченні різниці ($\Delta x_1 = x_2 - x_1, \ldots, \Delta x_{n-1} = x_n - x_{n-1}$). Синтаксис цієї команди:

- **diff(x) –** команда обчислює скінченні різниці, де **x** – одновимірний масив, а **diff(x) –**вектор різниць сусідніх елементів;
- **diff(x,n)** – команда обчислює скінченні різниці порядку **n**.

Приклад 1. Обчислити скінченні різниці незалежної та залежної змінних та перших похідних x, y, y', y'', якщо $y = \log x$.

Перелік команд для виконання цих операцій наданий нижче:

```
>> x=1:.1:2;
>> dx=diff(x);
>> y=log(x);
>> dy=diff(y);
>> d2y=diff(y,2);
```

Точне диференціювання функцій можна реалізувати за допомогою наступних команд:

- **diff(s)** – диференцює символьний вираз **s** по визначеній змінній;
- **diff(s,v)** – диференцює символьний вираз **s** по **v**;
- **diff (s,n)** або **diff(s,v,n) –** диференцює послідовно **n** разів символьний вираз **s.**

Приклад 2. Знайти першу та другу похідні функції $y = \frac{1}{2}\log(tg\frac{x}{2}) - \frac{1}{2}\frac{\cos x}{\sin^2 x}$.

Перелік команд для виконання вказаних операцій наведений нижче.

```
>> y=1/2*log(tan(x/2))–1/2*cos(x)/(sin(x))^2;
```

Знаходимо першу похідну:

```
>> Dy=diff(y)
Dy =
1/2*(1/2+1/2*tan(1/2*x)^2)/tan(1/2*x)+1/2/sin(x)+cos(x)^2/sin(x)^3
```

Знаходимо другу похідну:

```
>> D2y=diff(y,2)
D2y =
1/4+1/4*tan(1/2*x)^2–1/2*(1/2+1/2*tan(1/2*x)^2)^2/tan(1/2*x)^2–
5/2*cos(x)/sin(x)^2–3*cos(x)^3/sin(x)^4
```

Для отримання виразу вихідної функції виписуємо:

```
>> y=diff(y,0)
y =
1/2*log(tan(1/2*x))–1/2*cos(x)/sin(x)^2
```

Вправа 2. Частинні похідні та градієнт функції двох змінних.

Частинною похідною від функції двох змінних $z = f(x, y)$ за змінною x (позначається $z_x' = f_x'$ або $\frac{\partial z}{\partial x} = \frac{\partial f}{\partial x}$) називається границя відношення частинного прирісту функції $z = f(x, y)$ за змінною x до приросту цієї змінної Δx при умові $\Delta x \to 0$:

$$\frac{\partial z}{\partial x} = \lim_{\Delta x \to 0} \frac{\Delta_x z}{\Delta x} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x; y) - f(x; y)}{\Delta x},$$

похідна функції $z(x, y)$ по змінній x обчислена в припущенні, що інша змінна y фіксована [51].

Аналогічно визначається частинна похідна від функції $z = f(x, y)$ за змінною y (x вважається фіксованою):

$$\frac{\partial z}{\partial y} = \lim_{\Delta y \to 0} \frac{\Delta_y z}{\Delta y} = \lim_{\Delta y \to 0} \frac{f(x; y + \Delta y) - f(x, y)}{\Delta y}.$$

Приклад 1 Знайти частинні похідні функції $z = x^3 \sin y - y^2 \cos x$.

Визначимо символьні змінні та скористаємося командою **diff**:

```
>> syms x y z
>> z=sin(y)*x^3–y^2*cos(x);
```

Визначимо похідну по змінній **y**:

```
>> dy=diff(z,y)
dy =
cos(y)*x^3–2*y*cos(x)
```

Визначимо похідну по змінній **x**:

```
>> dx=diff(z,x)
dx =
3*sin(y)*x^2+y^2*sin(x)
```

Градієнтом $\mathrm{grad}\, u$ (або ∇u) функції u від кількох змінних називається вектор, координати якого є частинними похідними за відповідними незалежними змінними.

Таким чином, для функції двох змінних $u = f(x, y)$

$$\mathrm{grad}\, u(x, y) = \frac{\partial u}{\partial x} \cdot \vec{\boldsymbol{i}} + \frac{\partial u}{\partial y} \cdot \vec{\boldsymbol{j}} = \left(\frac{\partial u}{\partial x}; \frac{\partial u}{\partial y} \right),$$

а для функції трьох змінних $u = f(x, y, z)$

$$\mathrm{grad}\, u(x, y, z) = \frac{\partial u}{\partial x} \cdot \vec{\boldsymbol{i}} + \frac{\partial u}{\partial y} \cdot \vec{\boldsymbol{j}} + \frac{\partial u}{\partial z} \cdot \vec{\boldsymbol{k}} = \left(\frac{\partial u}{\partial x}; \frac{\partial u}{\partial y}; \frac{\partial u}{\partial z} \right).$$

Кажуть, що в області $G \subset \mathrm{R}^n$ (n=2, 3) задано векторне поле, якщо в кожній точці $P \in G$ заданий вектор $\vec{\boldsymbol{F}} = \vec{\boldsymbol{F}}(P)$ (тобто задана вектор-функція декількох змінних). Отже, будь-яка скалярна функція (скалярне поле) $u = u(P)$, для якої існує перша похідна в області G, може породжувати в цій області векторне поле $\vec{\boldsymbol{F}}(P) = \mathrm{grad}\, u(P)$.

Функція $u(P)$ називається потенціалом векторного поля $\vec{\boldsymbol{F}} = \mathrm{grad}\, u$, а векторне поле $\vec{\boldsymbol{F}}$ називається *потенціальним полем*.

Якщо функція $z(x, y)$ задана у вигляді масиву чисел (значень функції в визначених точках), то обчислити градієнт такої функції можна за допомогою команди **gradient**.

Приклад 2. Обчислити величину градієнта функції $z = \sin(x^2 - y^3)$ в визначених точках (вузлах сітки).

Відповідний масив координат вузлів (значень незалежних аргументів) нехай задається за допомогою команди **meshgrid.** Тоді послідовність команд для обчислення градієнту має вигляд:

```
>> [x y]=meshgrid(–1:.1:1,–1:.1:1);
>> z=sin(x.^2–y.^3);
>> [px,py]=gradient(z)
```

Для побудови тривимірного зображення скористаємось командою **surf**:

```
>> surf(z)
```

Якщо, крім вищенаведеного треба знайти чисельні значення похідних функції z, можна використати такий синтаксис команди **gradient**: **[px,py]=gradient(z,dx,dy)**, де **dx, dy** можуть бути:

- скалярними величинами, які дорівнюються крокам розбивки сітки по вісям x та y;
- векторами, компоненти яких є координати вузлів сітки при розбивці з перемінним кроком.

Для диференціального рівняння $y' = f(x, y)$, права частина якого безперервна в заданій області D, має місце геометрична інтерпретація, яка називається *полем напрямків*. Якщо через кожну точку (x, y) області D провести деякий відрізок $l(x, y)$ з кутовим коефіцієнтом $f(x, y)$, то вийде геометрична картина, яка і визначає поле напрямків: відрізок $l(x, y)$ являється дотичною в кожній точці $(x,\ y(x))$ інтегральної кривої $y = y(x)$ рівняння

Приклад 3. Обчислити та побудувати поле напрямків для функції $z = e^{x^2} - e^{(y^2 - x^2)}$.

Перелік команд наведений нижче:

```
>> [x,y]=meshgrid(–1:.1:1,–1:.1:1);
>> z=exp(x.^2)–exp(y.^2–x.^2);
>> [px,py]=gradient(z,pi/10,pi/10);
>> contour(z);
>> hold on;
>> quiver(px,py);
```

Для отримання вигляду поля напрямків використовуємо команду **surf:**

```
>> surf(z)
```

Вправа 3. Знаходження інтегралів від визначених функцій за допомогою підпакету **Simbolic Math**.

З появленням вищевказаного підпакету **Simbolic Math** стало можливим знаходити аналітічне значення деяких інтегралів.

Приклад 1. Знайти інтеграл від функції: $y = \frac{1+\sin x}{1+\cos x} e^x$.

Спочатку визначимо символьні змінні, потім виконаємо операцію інтегрування за допомогою команди **int**:

```
>> syms x y
>> y=(1+sin(x))*exp(x)/(1+cos(x)); iy=int(y)
iy =
exp(x)*tan(1/2*x)
```

Приклад 2. Обчислити визначений інтеграл $\int\limits_0^{\ln 2} sh^4 dx$.

Ця задача виконується за допомогою команди і**nt**:

```
>> int((sinh(x))^4,0,log(2))
ans =
   0.0402
```

У більш складному випадку подвійних інтегралів значна увага приділяється визначенню області інтегрування.

Якщо область інтегрування σ обмежена кривими $y_1 = \varphi_1(x),\ y_2 = \varphi_2(x)$, які кожна пряма, що паралельна вісі OY, перетинає не більш ніж у одній точці, то подвійний інтеграл по цій області може бути обчислений за допомогою процедури переходу до послідовного інтегрування:

$$\iint\limits_{(\sigma)} f(x,y)dxdy = \int\limits_a^b dx \int\limits_{\varphi_1(x)}^{\varphi_2(x)} f(x,y)dy .$$

Якщо область інтегрування σ обмежена кривими $x_1 = \varphi_1(y),\ x_2 = \varphi_2(y)$, які кожна пряма, що паралельна вісі OX, перетинає не більш ніж у одній точці, то подвійний інтеграл по цій області може бути обчислений за допомогою наступної процедури переходу до послідовного інтегрування:

$$\iint\limits_{(\sigma)} f(x,y)dxdy = \int\limits_a^b dy \int\limits_{\varphi_1(y)}^{\varphi_2(y)} f(x,y)dx .$$

Приклад 3. Обчислити інтеграл $\int\limits_{-3}^{3} dy \int\limits_{y^2-4}^{5} (x+2y)dx$:

Спочатку обчислюємо внутрішній інтеграл:

```
in1=int(x+2*y,y^2–4,5)
in1 =
25/2–1/2*(y^2-4)^2+2*y*(9–y^2)
```

Обчислюємо зовнішній інтеграл:

```
in2=int(in1,–3,3)
in2 =
252/5
```

Вправа 4. Чисельне інтегрування у системі **MATLAB**.

У системі **MATLAB** реалізовані різні методи чисельного інтегрування функцій. Одним з найбільш розповсюджених методів наближеного обчислення визначених інтегралів є метод трапецій. Для його реалізації в системі **MATLAB** існує команда **trapz.**

Приклад 1. Обчислити визначений інтеграл $\int\limits_{-\pi}^{\pi} \sin x^2 dx$ за допомогою методу трапецій.

Перелік команд процедури обчислення визначеного інтегралу за умови фіксованого кроку інтегрування $h = \dfrac{\pi}{10}$:

```
>> x=–pi:pi/10:pi;
>> y=sin(x.^2);
>> disp(trapz(x,y))
   1.4448
```

Якщо зменьшити крок інтегрування до $h = \dfrac{\pi}{1000}$, можна отримати більш точний результат:

```
>> x=–pi:pi/1000:pi;
>> y=sin(x.^2); disp(trapz(x,y))
   1.5453
```

Методи інтегрування більш високих порядків точністі у системі **MATLAB** реалізовані за допомогою таких команд, як **quad** (метод Симпсона), **quadl** (метод Гаусса-Лобатто), **quad8** (метод Ньютона-Котеса 8-го порядку).

Загальне звертання до команди **quad**: **[s,n]=quad(f,a,b,tol,trace,...)**, де **s** – значення інтегралу, **n** – кількість обчислень підінтегральної функції, **f** – функція, **[a, b]** – інтервал інтегрування, **tol** – відносна похібка (за замовчуванням вона дорівнює **1e-3**), **trace** – при ненульовому значенні виводяться послідовність етапів обчислення інтегралу.

Приклад 2. Обчислити визначений інтеграл $\int_0^1 \frac{e^{-x}}{x^3+1}dx$.

Для завдання підінтегральної функції створюємо **m**-файл:

```
function f=int1(x)
f=exp(–x)./(x.^3+1);
```

Обчислюємо інтеграл за допомогою різних команд інтегрування **quad**, **quad8**, **quadl**:

```
>> format long
>> [in1,n]=quad('int1',0,1,1e–6,1)
     9    0.0000000000    2.71580000e–001    0.2367423251
    11    0.2715800000    4.56840000e–001    0.2475552724
    13    0.2715800000    2.28420000e–001    0.1468823197
    15    0.5000000000    2.28420000e–001    0.1006731351
    17    0.7284200000    2.71580000e–001    0.0704552012
in1 =
   0.55475298116279
n =
    17
>> [in1,n]=quad8('int1',0,1,1e–6,1)
in1 =
   0.55475295813348
n =
    33
>> [in1,n]=quadl('int1',0,1,1e–6,1)
    18    0.0000000000    5.00000000e–001    0.5547529353
in1 =
   0.55475293530993
n =
    18
```

Якість обчислення виявляється в результаті порівняння отриманих значень інтегралу з точним результатом.

Для наближеного обчислення послідовного інтегралу у системі **MATLAB** використовується команда, яка має наступний синтаксис:

dblquad(f,xmin,xmax,ymin,ymax),

де **f –** підінтегральна функція**, xmin,xmax,ymin,ymax –** інтервали інтегрування відповідно по змінним x та y.

Приклад 3. Обчислити послідовний інтеграл $\int\limits_{-\frac{\pi}{2}}^{0} dx \int\limits_{0}^{\frac{\pi}{2}} (x^2 - y^3) dy$.

Скористаємося вищенаведеною командою **dblquad**, для якої спочатку задаємо інтервали інтегрування:

```
>> f=inline('x.^2–y.^3');
>> disp(dblquad(f,–pi/2,0,0,pi/2))
  –0.3614
```

Вправа 5. Розклад в степеневі ряди та наближене представлення функцій.

Функція $f(x)$ може бути представлена у вигляді степеного ряду за степенями $(x-a)$, а саме: $f(x) = a_0 + a_1(x-a) + a_2(x-a)^2 + ... + a_n(x-a)^n + ...$. Пошук коефіцієнтів $a_i, i = 0,1,2,...$. Такий ряд можна у наступному вигляді: $f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + ... + \frac{f^{(n)}(a)}{n!}(x-a)^n + ...$. Цей ряд носить назву *ряду Тейлора* для функції $f(x)$.

У випадку, коли $a = 0$, вихідний ряд набуває вигляду $f(0) + \frac{f'(0)}{1!}x + \frac{f''(0)}{2!}x^2 + ... + \frac{f^{(n)}(0)}{n!}x^n + ...$, який називається *рядом Маклорена* для функції $f(x)$.

Для того, щоб сума ряду збігалась до функції $f(x)$ необхідно і достатньо виконання вимоги: залишковий член $R_n(x) = f(x) - S_n(x)$, де $S_n(x)$ – сума n перших членів в ряду, повинен прагнути до нуля при нескінченному зростанні чинника n, $n \to \infty$, тобто $\lim\limits_{n\to\infty} R_n(x) = 0$.

У системі **MATLAB** надано декілька команд, що дозволяють представити задану функцію у вигляді ряду Тейлора:

- **taylor(f)** – розкладає функцію в ряд Маклорена (за замовчуванням призначається розкладання до 5-ої степені);
- **taylor(f,n)** – розкладає функцію в ряд Маклорена **n-1** степені;
- **taylor(f,a)** – розкладає функцію у ряд Тейлора в околі точки $x = a$;

- **taylor(f,x)** – розкладає функцію у ряд Маклорена у загальному виді до 5-ої степені (за замовчуванням) в околі точки **x.**

Приклад 1. Розкласти у степеневий ряд Тейлора функцію $f(x)=\sin x$.

Наведемо різні приклади розкладання функції $f(x)$ за допомогою вищевказаних команд:

```
>> taylor(sin(x))
ans =
x–1/6*x^3+1/120*x^5
```

Розкладання функції $f(x)$ у ряд Тейлора навколо точки **t**:

```
>> taylor(sin(x),t)
ans =
sin(t)+cos(t)*(x–t)–1/2*sin(t)*(x–t)^2–1/6*cos(t)*(x-
t)^3+1/24*sin(t)*(x–t)^4+1/120*cos(t)*(x–t)^5
```

Розкладання у ряд до 6-ої степені:

```
>> taylor(sin(x),6)
ans =
x–1/6*x^3+1/120*x^5
>> % Розкладання у ряд до 2-го ступеня навколо крапки 5:
>> taylor(sin(x),2,5)
ans =
sin(5)+cos(5)*(x–5)
```

Приклад 2. Знайти значення функції $f(x)=\sin x$ при $x=\pi$.

Процедура отримання необхідного значення функції може бути реалізована за наступною схемою. Спочатку розкладаємо функцію у ряд Маклорена до 6-ї степені:

```
>> sn1=taylor(sin(x),6)
sn1 =
x–1/6*x^3+1/120*x^5
```

Знайдемо границю отриманого ряду при $x \to \pi$:

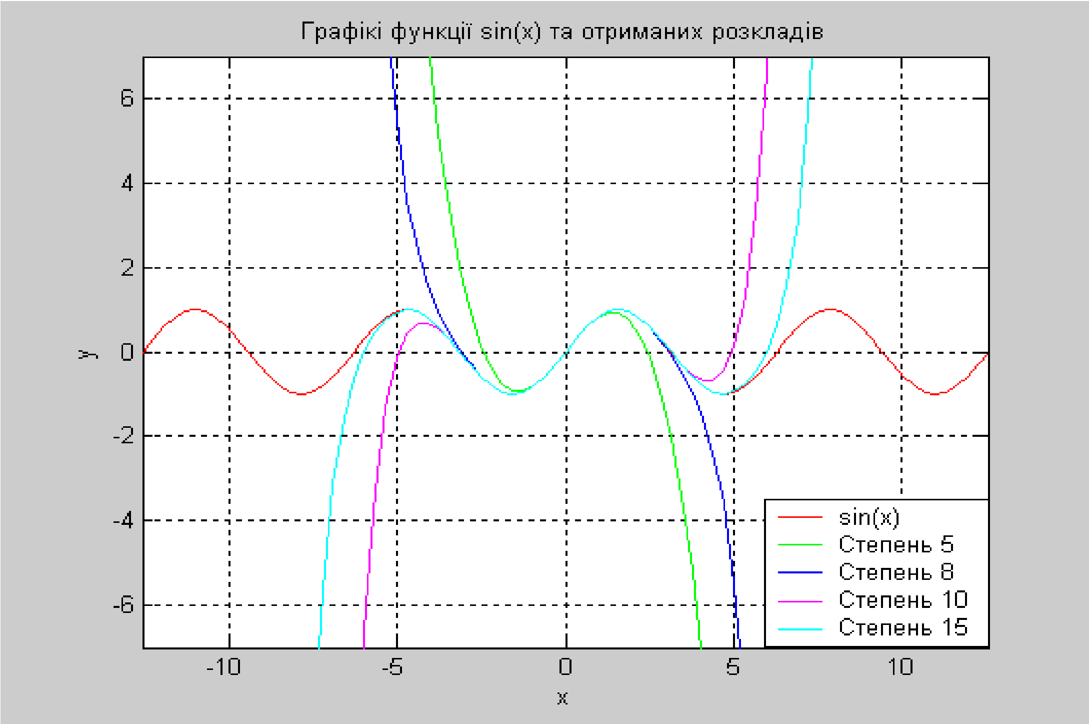
```
>> sn1=limit(x–1/6*x^3+1/120*x^5,pi)
sn1 =
pi–1/6*pi^3+1/120*pi^5
>> pi–1/6*pi^3+1/120*pi^5
ans =
   0.5240
```

З метою аналізу можливостей процедури побудуємо за допомогою команди **ezplot** з підпакету **Simbolic Math** графіки функції $f(x)=\sin x$ та графіки отриманих розкладів в ряди до різних степенів вихідної функції:

```
>> ezplot(sin(x))
>> hold on
>> ezplot(taylor(sin(x),5))
>> ezplot(taylor(sin(x),8))
>> ezplot(taylor(sin(x),10))
>> ezplot(taylor(sin(x),15));
>> grid on
```

Звернемо увагу на графікі, які надані на мал. 8.1. Вони відрізняються від графіків, які отримані за допомогою команди **ezplot**, лише оформленням.

З побудованих графіків (мал. 8.1) видно, що, чим більше степень розкладання функції в степеневий ряд, тим точніше наближене представлення до вихідної функції в околі нульового значення x.



Мал. 8.1 – Графікі функції $f(x) = \sin x$ та розкладів функції $f(x)$ в ряди Тейлора

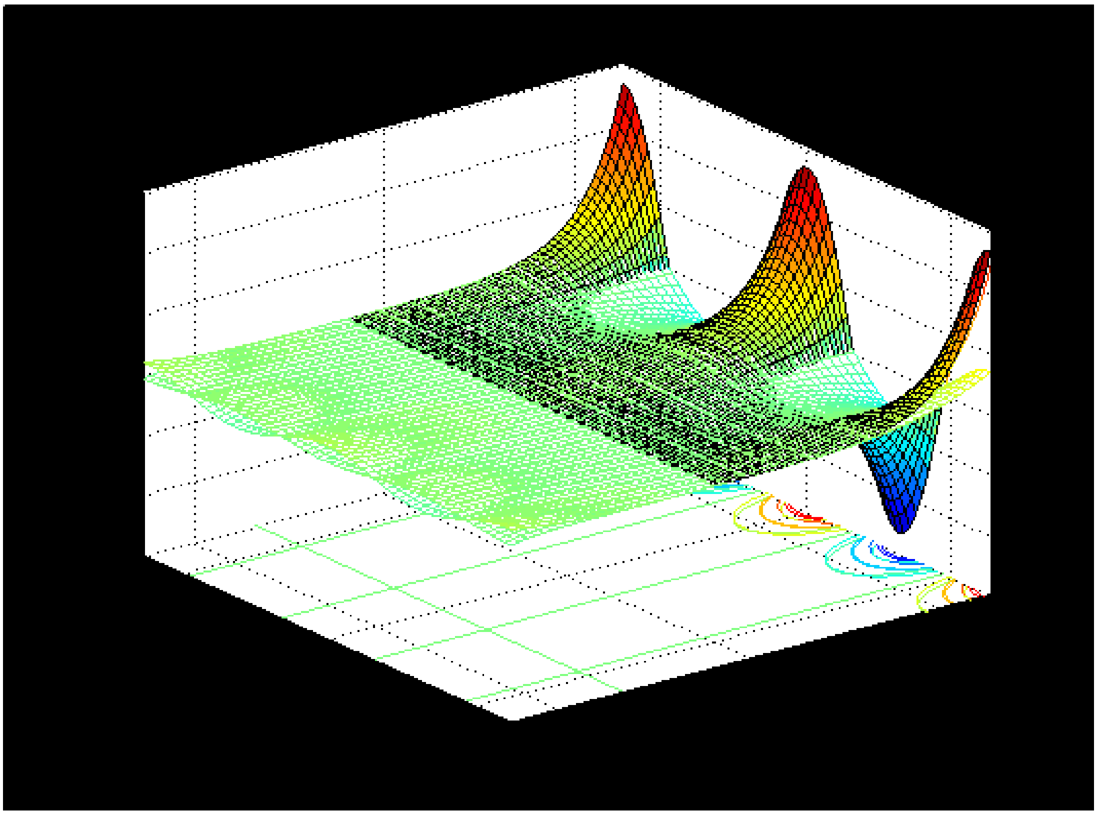
Приклад 3. Розкласти в степеневий ряд Тейлора функцію $f(x, y)$ двох аргументів: $f(x, y) = e^x \cos y$ та побудувати графіки вихідної функції і представити функції степеневим рядом.

Перелік команд, що допомогають виконати завдання:

```
>> taylor(exp(x)*cos(y))
```

```
ans =
cos(y)+cos(y)*x+1/2*cos(y)*x^2+1/6*cos(y)*x^3+1/24*cos(y)*x^
4+1/120*cos(y)*x^5
>> ezsurfc(exp(x)*cos(y));
>> hold on
>> ezmeshc(taylor(exp(x)*cos(y),5));
>> ezmeshc(taylor(exp(x)*cos(y),10));
```

В результаті отримуємо графіки функції $z(x, y)$ та розкладів цієї функції в ряди Тейлора різних порядків (мал. 8.2).



Мал. 8.2 – Графікі функції $f(x,y)=e^x \cos$ та функції $f(x, y)$ в ряди Тейлора