Лабораторна робота № 9.

Тема. Дослідження функцій. Апроксимація та інтерполяція даних.

Мета роботи: ознайомитись з процедурами дослідження функцій за допомогою команд у системі **MATLAB**; навчитись обчислювати екстремуми функцій, перевіряти функції на неперервність та знаходити точки розриву функцій; навчитись аналізувати експериментальні дані за допомогою засобів системи **MATLAB**.

Теоретичний мінімум

Загальна схема дослідження функції [30, с.471] має наступний вигляд.

1. Знаходимо область визначення D функції $f(x)$.
2. З'ясовуємо парність функції.
   - якщо $f(-x) = f(x)$, то функція $f(x)$ називається *парною*. Графік парної функції симетричний відносно вісі ординат (вісі OY).
   - якщо $f(-x) = f(-x)$, то функція $f(x)$ називається *непарною*. Графік непарної функції симетричний відносно початку координат.
3. З'ясовуємо періодичність функції.

Якщо $f(x+T) = f(x)$ при деякому $T > 0$, то функція $y = f(x)$ називається періодичною. Графік періодичної функції має одну і ту ж форму на кожному з відрізків $...[-2T;-T], [-T, 0],[0, T], [T, 2T],...$ Тому досить побудувати графік на якому-небудь одному такому відрізку і потім відтворити отриману криву на решті відрізків

4. Знаходимо точки максимуму і мінімуму функції та інтервали зростання і спадання функції (інтервали монотонності). Для цього:
   - обчислюємо похідну $f'(x)$ і знаходимо так звані *критичні (стаціонарні точки) точки* функції, тобто точки, в яких похідна $f'(x) = 0$, похідна прямує до $\pm\infty$ або не існує;
   - визначаємо знак похідної і знаходимо інтервали зростання і спадання функції: якщо $f'(x) > 0$, то функція зростає, якщо $f'(x) < 0$, то функція спадає;
   - якщо похідна змінює знак під час переходу через критичну точку $x_0 \in D$, то x_0 є точкою екстремуму:
   - якщо похідна міняє знак з «мінуса» на «плюс», то x_0 є точкою мінімуму, якщо ж з «плюса» на «мінус», то x_0 є точкою максимуму.
   - якщо похідна зберігає знак під час переходу через критичну точку, то в цій точці екстремум не визначається.

5. Знаходимо точки перегину функції і інтервали опуклості і угнутості функції. Для цього:

- обчислюємо другу похідну $f''(x)$ і знаходимо точки, що належать області визначення функції, в яких похідна $f''(x)=0$ або прямує до $\pm\infty$;
- за допомогою знаку другої похідної функції, знаходимо інтервали опуклості і угнутості: якщо друга похідна функції міняє знак під час переходу через точку $x_0 \in D$, в якій $f''(x)=0,\pm\infty$ або не існує, то x_0 – точка перегину.

6. Знаходимо асимптоти функції $y=f(x)$.

Вертикальні асимптоти: знаходимо односторонні границі в крапках $x=x_k$ $\lim\limits_{x\to x_k-0} f(x)$ і/або $\lim\limits_{x\to x_k+0} f(x)$. Якщо хоч би одна з цих границь нескінченна, то $x=x_k$ – вертикальна асимптота графіка функції $y=f(x)$.

Похилі асимптоти: якщо існують кінцеві границі

$$\lim_{x\to\pm\infty}\frac{f(x)}{x}=k \ \text{ i } \ \lim_{x\to\pm\infty}[f(x)-kx]=b,$$

то пряма $y=kx+b$ визначає похилу асимптоту графіка функції $y=f(x)$ (якщо $k=0, b=\lim\limits_{x\to\pm\infty} f(x)$, то $y=b$ визначає горизонтальну асимптоту).

Зауваження 1. Асимптоти при $x\to+\infty$ і $x\to-\infty$ можуть бути різними.

Зауваження 2. При необхідності можна знайти точки перетину кривої з вісями координат і визначити значення функції в додаткових точках.

Вправа 1. Здійснити повне дослідження функції $y=\dfrac{x^2}{(x-1)^2}$.

1. Знаходження області визначення. По-перше можна знайти значення змінної x, при якій функція невизначена (тобто знаменник вихідної функції обертається в нуль). Для цього згнаходимо розв'язок рівняння $(x-1)^2=0$ за допомогою команди **solve**:

```
>> syms x y; solve((x–1)^2)
ans =
[ 1]
[ 1]
```

Таким чином, область визначення: $D(y)=(-\infty,1)\cup(1,+\infty)$.

2. Визначення парності або непарності функції. Для чого підставляємо у вихідну функцію $f(x)$ значення аргументу $(-x)$:

```
>> y=(–x)^2/(–x+1)^2
y =
x^2/(–x+1)^2
```

Функція не є парною, ні непарною, тобто її графік не є симетричним не відносно вісі Ox, не відносно початку координат.
3. Функція не є періодичною, що витікає з факту наявності скінченої кількості нулів у даній функції, тобто точок x_k, в який $y(x_k)=0$.
4. Знайдемо точки максимуму і мінімуму функції та інтервали зростання і спадання. Для цього залучимо команди:

- **diff** – пошуку похідної;
- **fzero** – обчислення нулів функції;
- **fminbnd** – знаходження локального мінімуму.

Для встановлення інтервалів зростання та спадання функції, необхідно обчислити першу похідну і знайти її нулі:

```
>> dy=diff(x^2/(x–1)^2)
dy =
2*x/(x–1)^2-2*x^2/(x–1)^3
>> solve(dy)
ans =
0
```

Очевидно, що $y'=0$ при не існує при $x=1$. Інформацію про поведінку функції поміщаємо в таблицю:

x	$(-\infty,\ 0)$	0	$(0,\ 1)$	1	$(1,\ +\infty)$
y'	–	0	+	не існує	–
y	↘	0	↗	не існує	↘

Функція спадає на інтервалі $(-\infty,\ 0),\ (1,\ +\infty)$, тобто $x\in(-\infty,\ 0)\cup(1,\ +\infty)$.

Знаходимо локальний мінімум функції:

```
>> [xmin]=fminbnd('x^2/(x–1)^2',-1,1)
Warning: Divide by zero.
(Type "warning off MATLAB:divideByZero" to suppress this warning.)
> In C:\MATLAB6p5\toolbox\MATLAB\funfun\inlineeval.m at line 13
  In C:\MATLAB6p5\toolbox\MATLAB\funfun\@inline\feval.m at line 34
  In C:\MATLAB6p5\toolbox\MATLAB\funfun\fminbnd.m at line 208
>> xmin =
  8.6043e-006 %
```

Функція зростає при $x\in(0,\ 1)$, висновок: $(0,\ 0)$ – точка мінімуму.
5. Опуклість вверх та опуклість вниз кривої. Для встановлення вказаних форм кривої треба відшукати другу похідну та знайти її нулі:

```
>> ddy=diff(2*x/(x–1)^2–2*x^2/(x–1)^3)
ddy =
2/(x–1)^2–8*x/(x–1)^3+6*x^2/(x–1)^4
>> [x]=solve(ddy)
x =
–1/2
```

```
>> x=–0.5;y=x^2/(x–1)^2
y =
   0.1111 % (1/9)
```

Тобто $y''=0$ при $x=-0.5$ і y'' не існує при $x=1$.

Коли $x\in(-\infty;-0.5)$ і $y''<0$, то крива опукла вверх.

Коли $x\in(-0.5,1)$ і $y''>0$, то крива опукла вниз.

Коли $x\in(1,+\infty)$ і $y''>0$, то крива опукла вниз.

Точка $\left(-0.5,\ \frac{1}{9}\right)$, в якій крива змінює форми опуклості є точкою перегину.

6. Встановлення асимптот до графіку вихідної функції. Можливо встановити три види асимптот – вертикальні, похілі та горизонтальні. Проаналізуємо поведінку функції на границі області визначення. В точці $x=1$ маємо границю $\lim\limits_{x\to1}\frac{x^2}{(x-1)^2}=\infty$:

Висновок: вертикальна асимптота, її рівняння: $x=1$.

Для встановлення форми представлення похилої асимптоти треба визначити коефіцієнт k та сталої b прямої лінії $y=kx+b$. Дані чинники знаходяться за формулами:

$$\lim_{x\to\pm\infty}\frac{f(x)}{x}=k,\ \lim_{x\to\pm\infty}[f(x)-kx]=b.$$

Відповідні команди системи **MATLAB:**

```
>> syms x; limit(x/(x–1)^2,inf)
ans =
0
>> limit(x^2/(x–1)^2,inf)
ans =
1
```

Висновок: рівняння $y=1$ задає похилу (горизонтальну) асимптоту.

7. Будуємо графік функції та графікі знайдених асимптот в єдиному графічному вікні:

```
>> syms x y; ezplot(x^2/(x–1)^2)
>> ezplot(x^2/(x–1)^2);grid on;hold on
>> line([–6 6],[1 1]); line([1 1],[–.5 6.5]);
```

Треба звернути увагу на те, що побудова графіків функцій у системі **MATLAB** не є проблемою. Візуалізація графіків не тількі може спростити задачу пошуку нулів функції або точок екстремумів, а допомогти в пошуку інтервалів знаходження визначених точок та оцінити поведінку функції з загальних точок зору. Сучасні можливості дослідження функцій засобами системи **MATLAB** дозволяють розглядати і більш складні задачі ніж задачі про дослідження функцій.

Вправа 2. Знаходження найбільшого та найменьшого значень функції.

Неперервна на відрізку $[a,\ b]$ функція досягає на цьому відрізку свого найменьшого і найбільшого значення або в точках інтервалу $[a,\ b]$, або в кінцевих точках відрізка: $x = a,\ x = b$.

Приклад 1. Знайдемо найбільше та наіменьше значення функції $y(x) = 3x^4 - 20x^3 - 36x^2 + 5$ на відрізку [–1; 1].

Вихідна функція визначена і неперервна на всій чисельній вісі. Знайдемо критичні (екстремальні) точки, що є точками інтервалу $[a,\ b]$, в яких похідна y' функції $y(x)$ дорівнює нулю або не існує. Визначеємо похідну функції:

```
>> syms x y
yp=diff('3*x^4–20*x^3-36*x^2+5')
yp =
12*x^3–60*x^2–72*x
```

Оскільки похідна y' функції $y(x)$ визначена (існує) в усіх точках, то залишається знайти її нулі, для чого розв'язуємо рівняння $12x^3 - 60x^2 - 72x = 0$ за допомогою команди **solve**:

```
>> [x]=solve('12*x^3–60*x^2–72*x')
x =
[  0]
[ –1]
[  6]
```

З трьох отриманих коренів цього рівняння $x_1 = 0,\ x_2 = -1,\ x_3 = 6$ тільки перші два належать заданому відрізку [–1; 1]. Обчислюємо значення функції в внутрішніх точках відрізка і на кінцях відрізку:

```
>> x=[–1 0 1 ];
>> y=3.*x.^4–20.*x.^3–36.*x.^2+5
y =
   –8    5   –48
```

Порівняння отриманих значень функції $y(x)$ дозволяє знайти максимальне і мінімальне значення функції $y(x)$:

```
>> max(y)
ans =
    5
>> min(y)
ans =
  –48
```

Вправа 3. Методи обробки експериментальних даних у системі **MATLAB**.

Для аналізу експериментальних даних, які надані у вигляді таблиць і задають залежність одних фізичних величин від інших, використовують такі методи, як апроксимація, інтерполяція та згладжування.

Як відомо, в научних та інженерних розрахунках часто приходиться оперувати наборами значень, які були получені експериментальним засобом або методом випадкової виборки. Як правіло, на підставі цих наборів потрібно побудувати функцію, яка би з високою точністю описувала вихідний набір значень. Така задача називається задачою апроксимації даних. Інтерполяція – це різновид апроксимації, при якій графік функції, що буде побудований, проходить точно через задані точки вихідних даних [51, с.263].

Припустимо, що на числовій вісі задані точки $x_1 < x_2 < ... < x_n$, яким відповідають значення функціїї $y_1, y_2, ..., y_n$, де кожне x_i та y_i – дійсні числа, Завдання одновимірної апроксимації полягає у тому, щоб побудувати просту обчислювальну функцію, яка наближається цих даних. Залежно від класу апроксимуючих функцій і вимог до способу апроксимації виникають різні постановки завдань.

Нехай є n значень x_i, кожному з яких відповідає своє значення y_i. Потрібно знайти таку функцію $f(x)$, що: $f(x_i) = y_i$, $i = 1, ..., n$, точки x_i називають вузлами інтерполяції, пари (x_i, y_i) називають точками даних, різницю між «сусідніми» значеннями $x_i - x_{i-1}$ називають *кроком*, функцію $f(x)$ є інтерполюючою функцією або *інтерполянтом*.

На практиці в якості інтерполюючої функції $f(x)$ часто використовуються алгебраїчні поліноми різного вигляду, оскільки поліноми легко обчислювати, диференціювати і інтегрувати. При цьому інтерполяція носить назву поліноміальної [33, с.119].

У системі **MATLAB** визначена команда апроксимації даних поліномами за методом найменьших квадратів **polifit(x,y,n),** де

- **x –** вектор абсцис функції, що апроксимується;
- **y –** вектор ординат функції, що апроксимується;
- **n –** степень поліному, за допомогою якого апроксимуємо задану функцію.

Команда **polifit** формує коефіцієнти поліному $P_n(x)$ степені n, який з наіменьшею середньоквадратичною похибкою апроксимує вихідну функцію $y(x)$.

Приклад 1. Здійснити обробку експериментальних даних, застосовуючи поліноми другої та четвертої степенів.

Задаємо координати вузлів **x** та **y**:

```
>> x=0:.1:.5;y=[.25 .15 .05 .1 .12 .2];
>> P=polyfit(x,y,2);
```

P =

2.4643 –1.3150 0.2479

Для обчислення значень функції **yy** використовуємо команду **polyval**:

```
>> yy=polyval(P,x);
```

Побудуємо графік:

```
>> plot(x,y,'or',x,yy,'–b');grid;axis([0 0.5 –0.05 0.3]);hold on
```

Зверніть увагу на застосування команди **hold**, яка забезпечує відтворення графіків функцій в одному графічному вікні. Тепер задамо апроксимуючий поліном четвертої степені:

```
>> P=polyfit(x,y,4)
P =
 –12.5000   10.2778    0.2917   –1.2623    0.2524
>> yy=polyval(P,x);
```

Графік поліному четвертої степені будуємо в тому ж вікні, що і графік поліному другої степені:

```
>> plot(x,yy,'-m');axis([0 0.5 –0.05 0.3]);hold off
```

Результати обчислень можна порівняти візуально. Але, не зважаючи на те, що збільшується степень поліному, якість наближення може не покращуватися.

Вправа 4. Одновимірна апроксимація даних. Засоби апроксимації.

Подальше підвищення якості апроксимації експериментальних даних можна досягти і у випадку використання поліномів невисокої (зазвичай 3-ої) степені, при якому наближаються локально розташовані дані, окремо на кожному частковому інтервалі між сусідніми вузлами. При цьому такі поліноми обчислюються так, щоб не тільки їх значення збігалися з координатами вузлових точок, але щоб у вузлових точках похідні першого і другого порядків були непеперервні. Така поведінка характерна для гнучкої металевої лінійки, яка закріплюється у вузлових точках (така лінійка називається сплайном (**spline)**), Відповідний вид інтерполяції даних теж отримав назву сплайн-інтерполяції. Для демонстрації можливостей сплайн-інтерполяції розглянемо інтерполяцією, яка заснована на використанні так званих кубічних сплайнів [37].

Інтерполяція кубічними сплайнами у системі **MATLAB** за допомогою виконується команди **spline: yy=spline(x,y,xx),** де

- **x –** вектор абсцис функції, що апроксимується;
- **y –** вектор ординат функції, що апроксимується;
- **xx –** вектор абсцис контрольних точок, в яких обчислюються значення апроксимаційних поліномів; відповідні значення поліномів записуються у вектор **yy.**

Приклад 1. Виконати інтерполяцію кубічними сплайнами функції $y(x) = \sin x^2 - x$ на відрізку $x \in [0, 10]$ (шаг інтерполяції дорівнює одиниці).

Послідовність команд для виконання інтерполяції наведена нижче:

```
>> x=0:10;y=sin(x.^2)–x; xx=0:.1:10;yy=spline(x,y,xx);
>> plot(x,y,'b',xx,yy,'r');grid
```

Зверніть увагу, що у системі **MATLAB** існує команда **interp1**, яка дозволяє задати вид інтерполяції: **yy = interp1(x,y,xx,method)**, де:

- **x –** вектор абсцис функції, що апроксимується,
- **y –** вектор ординат функції, що апроксимується,
- **xx –** вектор абсциси контрольних точок, в яких обчислюються значення апроксимуючих поліномів; відповідні значення поліномів записуються у вектор **yy.**
- **method –** вид апроксимації, назва якого задається у вигляді рядка символів (насправді досить вказати тільки перший символ).

Команда **interp1** дозволяє використовувати наступні методи інтерполяції:

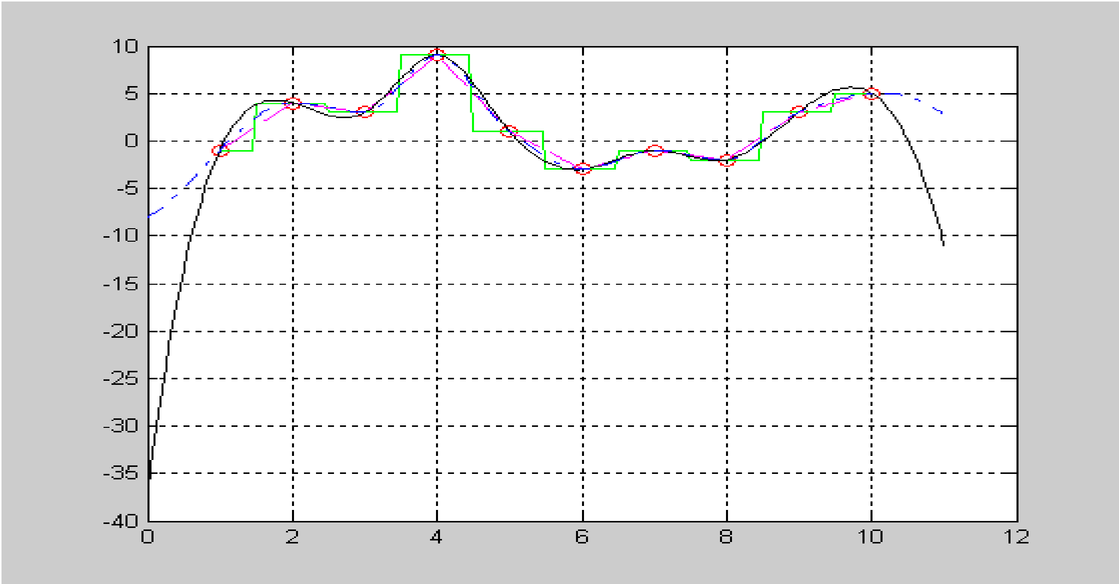
- **nearest –** апроксимація кусковими поліномами нульового ступеня (сходинками): для будь-якого проміжного значення **xx** знаходиться найближче табличне значення **x** і як у випадку формування вектора **yy** відшукується відповідне табличне значення **y;**
- **linear –** апроксимація кусковими поліномами 1-ої степені (ламаними);
- **spline –** апроксимація кусковими поліномами 3-ої степені (сплайнами);
- **pchip** або **cubic –** апроксимація кусковими поліномами Ерміта 3-ої степені.

Якщо параметр **method** команди **interp1** відсутній, за замовченням використовується команда **linear**, що виконує апроксимацію кусковими поліномами 1-ої степені.

Приклад 2. Виконати інтерполяцію експериментальних даних за допомогою різних видів інтерполяції, якщо вихідні значення даних задані у вигляді векторів $x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$, $y = [-1, 4, 3, 9, 1, -3, -1, -2, 3, 5]$.

Послідовність команд для здійснення інтерполяції наступна (див. мал. 9.1):

```
>> x=[1 2 3 4 5 6 7 8 9 10]; y=[–1 4 3 9 1 –3 –1 –2 3 5];
>> xx=.0:.05:11;
>> yy1=interp1(x,y,xx,'nearest');
>> yy2=interp1(x,y,xx,'linear');
>> yy3=interp1(x,y,xx,'spline');
>> yy4=interp1(x,y,xx,'cubic');
>> plot(x,y,'or');grid on;hold on;
>> plot(xx,yy1,'g-',xx,yy2,'m--',xx,yy3,'k-',xx,yy4,'b-.');
```



Мал. 9.1 – Інтерполяція даних за допомогою різних видів інтерполяції

Термін інтерполяція визначає, зазвичай, обчислення значень функції $y(x)$ і в проміжках між вузловими точками.

Для апроксимації періодичних (і особливо гладких періодичних) функцій використовується підхід на основі застосування тригонометричного ряду Фур'є. Для цієї задачі застосовується спеціальна команда: **interpft(x, n),** де **y** – одновимірний масив (вектор ординат) значень функції, який визначений в **n** рівномірно розташованих точках.

Зауважимо, що всі методи інтерполяції вимагають, щоб значення елементів вектора **x** змінювалися монотонно.

Приклад 3. Виконати інтерполяцію функції $y(x)=\cos^3 x$ на відрізку $x \in [0, 10]$.

Опис команд, що розв'язує дану задачу, має наступний вигляд:

```
>> x=0:10;y=cos(x).^3;
>> x1=0:.1:10;y1=interpft(y,101);
>> x2=0:.01:10;y2=cos(x2).^3;
>> plot(x,y,'r');hold on;plot(x1,y1,'g',x2,y2,'b');
```

Вправа 5. Двовимірна інтерполяція.

Двувимірна інтерполяція здебільшого складніша, ніж одновимірна, яка була розглянута раніше. Сутність такої інтерполяції така ж сама – знайти проміжні точки деякої залежності $z(x, y)$ при завдання вузлових точок в просторі.

Для інтерполяції даних, які задають деяку поверхню на двувимірній сітці, використовується команда **interp2(x1,x2,y,z1,z2,'method').** Параметри цієї команди також повинні змінюватись монотонно і мати формат команди **meshgrid (**ця команда визначає сітку**)**. У випадку відсутності параметру **method** в команді **interp2** за замовуванням здійсюється лінійна інтерполяція даних. При цьому остаточний масив визначається на більш дрібнішій сітці.

Приклад 1. Здійснити двовимірну інтерполяцію функції . $z + y = x^2$.

Задача виконується наступним чином:

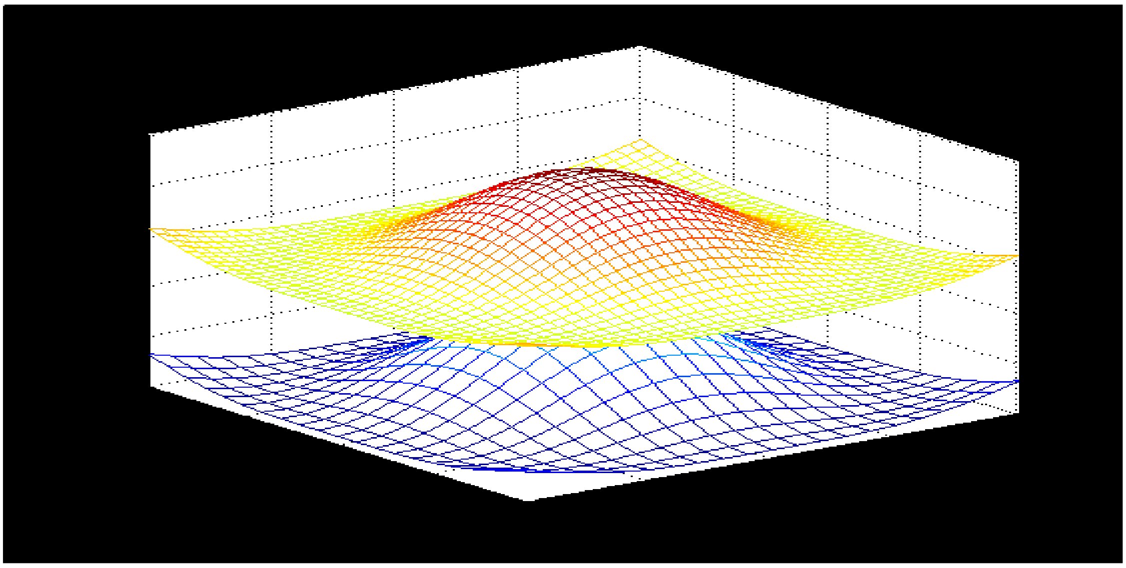
```
>> [x,y]=meshgrid(–2:0.1:2);
z=x.^2–y;
[x1,y1]=meshgrid(–2:0.05:2);
z1=interp2(x,y,z,x1,y1);
mesh(x,y,z),hold on,mesh(x1,y1,z1+2),hold off;
```

Зверніть увагу, що в останньому рядку в команді **mesh(x1,y1,z1+2)** до вихідного параметру було додане число 2. Це зроблено для кращої візуалізації графічної поверхні, що отримується, і необхідності відокремліти знов отриману поверхню від віхидної.

Приклад 2. Виконати двовимірну інтерполяцію функції $z(x, y) = (-x^2 - y^2)e^{-x^2-y^2}$.

Застосуємо наступну послідовність команд:

```
[x,y]=meshgrid(-1:0.1:1);
z=exp(–x.^2–y.^2).*(–x.^2–y.^2);
[x1,y1]=meshgrid(–1:0.05:1);
z1=interp2(x,y,z,x1,y1);
>> mesh(x,y,z),hold on,mesh(x1,y1,z1+.5),hold off;
```



Мал. 9.2 Двовимірна інтерполяція даних