Розділ 10. Застосування Maple до розв'язування задач лінійної алгебри

§ 10.1. Вектори та операції з ними

Основна частина команд для розв'язування задач лінійної алгебри міститься у бібліотеці `linalg`, тому перед розв'язуванням задач з векторами чи матрицями потрібно завантажити цю бібліотеку командою

```
> with(linalg):
```

Нагадаємо, що вектор у Maple можна задати за допомогою команд `vector` або `array` зі стандартної бібліотеки (стор. 41). Наприклад,

```
> v:=vector([2,0,-1]); w:=array([a,b,c]);
```

$$v := \begin{bmatrix} 2 & 0 & -1 \end{bmatrix}$$
$$w := \begin{bmatrix} a & b & c \end{bmatrix}$$

Координату вже визначеного вектора v отримуємо, використавши запис `v[i]`, де i – номер координати, а після виконання інструкції `v[i]:=a` i-му елементу вектора v буде присвоєно нове значення a. Наприклад,

```
> v[1];
```

$$2$$

```
> v[1]:=5: v[1];
```

$$5$$

Нагадаємо (§ 5.2), що вектор можна перетворити у список, а список у вектор за допомогою команд `convert(`*vector*`,list)` і `convert(`*list*`,vector)` відповідно. Наприклад, команда

```
> s:=convert(w,list);
```

перетворює вектор w у список:

$$s := [a, b, c]$$

Додати два вектори v і w однакової розмірності можна за допомогою команд `evalm(`v+w`)` або `matadd(`v, w`)`. Наприклад,

```
> v:=vector([1,2,3]): w:=vector([3,2,1]):
> s:=evalm(v+w);
```

$$s := \begin{bmatrix} 4 & 4 & 4 \end{bmatrix}$$

Команду `evalm` можна використовувати для множення вектора на число, віднімання векторів, для знаходження лінійної комбінації двох або більшої кількості векторів. Наприклад,

```
> v:=vector([0,1,2]): w:=vector([-3,-2,1]):
> evalm(0.5*v);
```

$$\begin{bmatrix} 0 & 0.5 & 1.0 \end{bmatrix}$$

```
> z:=evalm(2*v-3*w);
```

$$z := \begin{bmatrix} 9 & 8 & 1 \end{bmatrix}$$

Для знаходження скалярного добутку (v, w) векторів v і w використовується команда `dotprod(`v, w`)`, а для знаходження векторного добутку $[v, w]$ тих самих векторів – команда `crossprod(`v, w`)`. Кут α між векторами v і w легко знайти за допомогою команди `angle(`v, w`)`. Приклади:

```
> v:=vector([2,1,3]): w:=vector([1,4,-2]):
> (v,w)=dotprod(v,w); [v,w]=crossprod(v,w);
```

$$(v, w) = 0$$

$$[v, w] = \begin{bmatrix} -14 & 7 & 7 \end{bmatrix}$$

```
> alpha:=angle(v,w);
```

$$\alpha := \frac{1}{2}\pi$$

Довжину $|v|$ вектора v можна знайти командою `norm(`$v, 2$`)`, а пронормувати цей вектор (знайти орт $|v|^{-1}v$) – командою `normalize`(v). Наприклад,

```
> v:=vector([0,3,4]): n:=norm(v,2);
```

$$n := 5$$

```
> n1:=normalize(v);
```

$$n1 := \begin{bmatrix} 0 & \frac{3}{5} & \frac{4}{5} \end{bmatrix}$$

Команда `norm`(v, p) дозволяє знайти норму вектора v за формулою

$$\|v\|_p = \left(\sum_{i=1}^{n} |v_i|^p \right)^{1/p},$$

де n – розмірність вектора, v_i – його i-й елемент, а p – натуральне число. Якщо пропустити параметр p, то норма вектора v буде обчислюватись за формулою $\|v\| = \max\limits_{1 \leqslant i \leqslant n} |v_i|$. Наприклад,

```
> norm(v,1); norm(v);
```

$$7$$
$$4$$

Для знаходження базису сукупності векторів v_1, …, v_n використовується команда `basis`([$v_1, \ldots, v_n$]). Команда `GramSchmidt`([$v_1, \ldots, v_n$]) ортогоналізує сукупність лінійно незалежних векторів v_1, …, v_n за процедурою Грамма – Шмідта. Виділимо, наприклад, з сукупності векторів $v_1 = (1, 2, 2, -1)$, $v_2 = (1, 1, 5, 2)$, $v_3 = (2, 2, 3, 1)$, $v_4 = (1, 0, 0, 1)$, $v_5 = (5, 1, 0, 0)$ базис і ортогоналізуємо його:

```
> v1:=vector([1,2,2,-1]): v2:=vector([1,1,5,2]):
> v3:=vector([2,2,3,1]): v4:=vector([1,0,0,1]):
> v5:=vector([5,1,0,0]): basis([v1,v2,v3,v4,v5]);
```

$$[v1, v2, v3, v4]$$