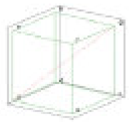


## Table of Contents



# ch02 sec2.2 向量的加法操作

**Outcomes**

- A 计算向量的和与差
- B 使用向量加法法则证明向量表达式等价

当两个向量表示为坐标向量形式.如:$u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}$ 和 $v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$, 加法可以定义为:

**Definition**

$$u + v = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} u_1 + v_1 \\ u_2 + v_2 \\ \vdots \\ u_n + v_n \end{bmatrix}$$

两个向量的元素数量要相同, 执行加法时对位相加

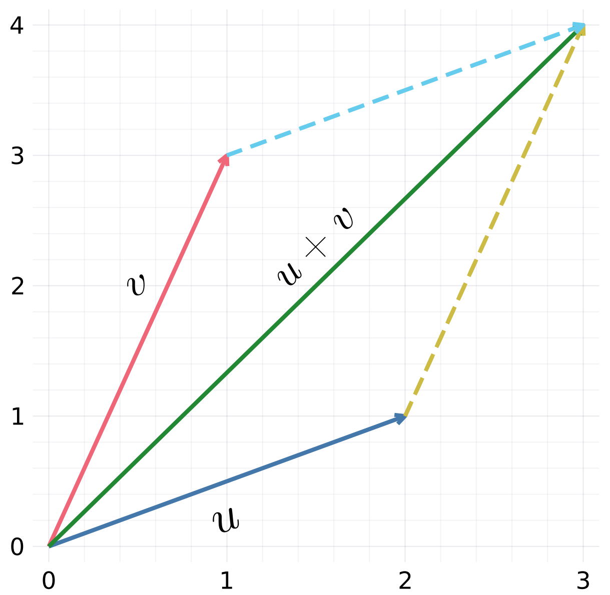
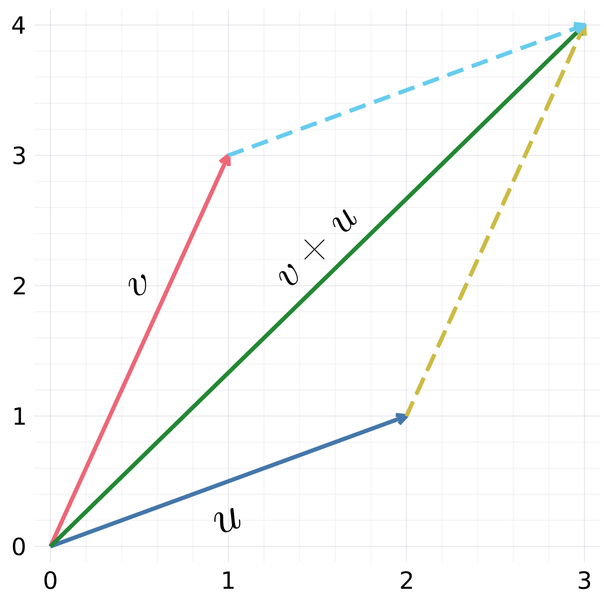
```
u = 4×1 adjoint(::Matrix{Int64}) with eltype Int64:
     1
     2
     3
     4
```

```
v = 4×1 adjoint(::Matrix{Int64}) with eltype Int64:
     4
     5
     6
     7

sum = 4×1 Matrix{Int64}:
        5
        7
        9
       11
```

## 在坐标系中表示向量的加法.

如果向量表示为坐标向量, 我们在上一节说过,所有的向量的起点都在原点, 坐标向量之间的加法累积了在各个坐标轴方向上的变化.

用二维坐标向量来说明这个问题,如图

| s | u+v | v+u |
|:--|:-------------:|:---------:|
| |  |  |

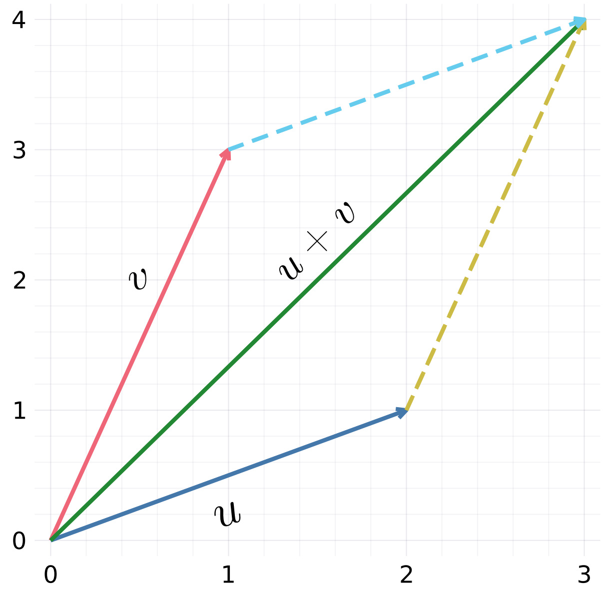


两个坐标向量以及他们的和组成了一个平行四边形.

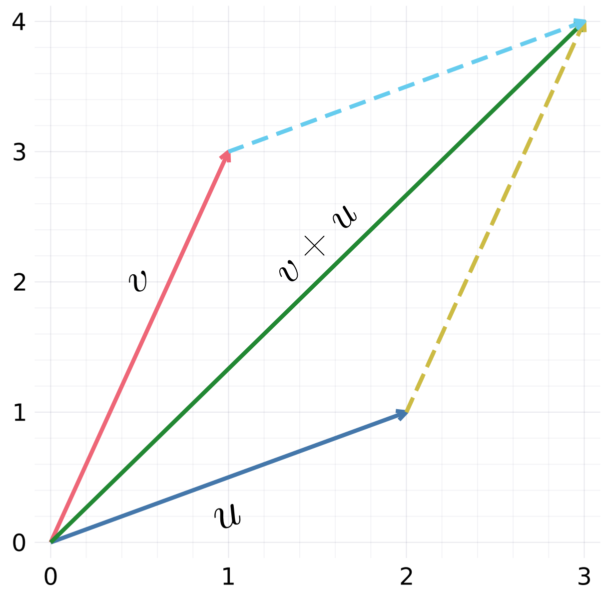
对角线是两个坐标向量的和, 方向为从原点指向两个向量对位元素和组成的坐标向量, 坐标向量和符合交换律, 交换位置, 不改变结果

```
md"""
## 在坐标系中表示向量的加法.

如果向量表示为坐标向量, 我们在上一节说过,所有的向量的起点都在原点, 坐标向量之间的加法累积了在各个坐标轴方向上的变化.

用二维坐标向量来说明这个问题,如图

 s   | u+v | v+u
:--| :-------------: | :---------:
 | $(store["vec_sum1"]) | $(store["vec_sum2"])


两个坐标向量以及他们的和组成了一个平行四边形.


对角线是两个坐标向量的和, 方向为从原点指向两个向量对位元素和组成的坐标向量, 坐标向量和符合交换律, 交换位置, 不改变结果



```

```
"""
```




```
let
    zero=[0,0]
    u=[2,1]
    v=[1,3]
    sum1=u+v
    sum2=v+v
    ann=[
        (1,0.2,text(L"u",pointsize=16,rotation=20)),
        (0.5,2,text(L"v",pointsize=14,rotation=20)),
        (1.5,2.3,text(L"u+v",pointsize=14,rotation=45))
    ]
    p1=vec_plot(zero,u)
    p2=vec_plot!(zero,v)
    p3=vec_plot!(zero,sum1)
    p4=vec_plot!(u,sum1,:dash)
    p5=vec_plot!(v,sum1,:dash)
    p6=plot!(ann=ann,size=(320,320))

    save("vec_sum1",p6)

end
```

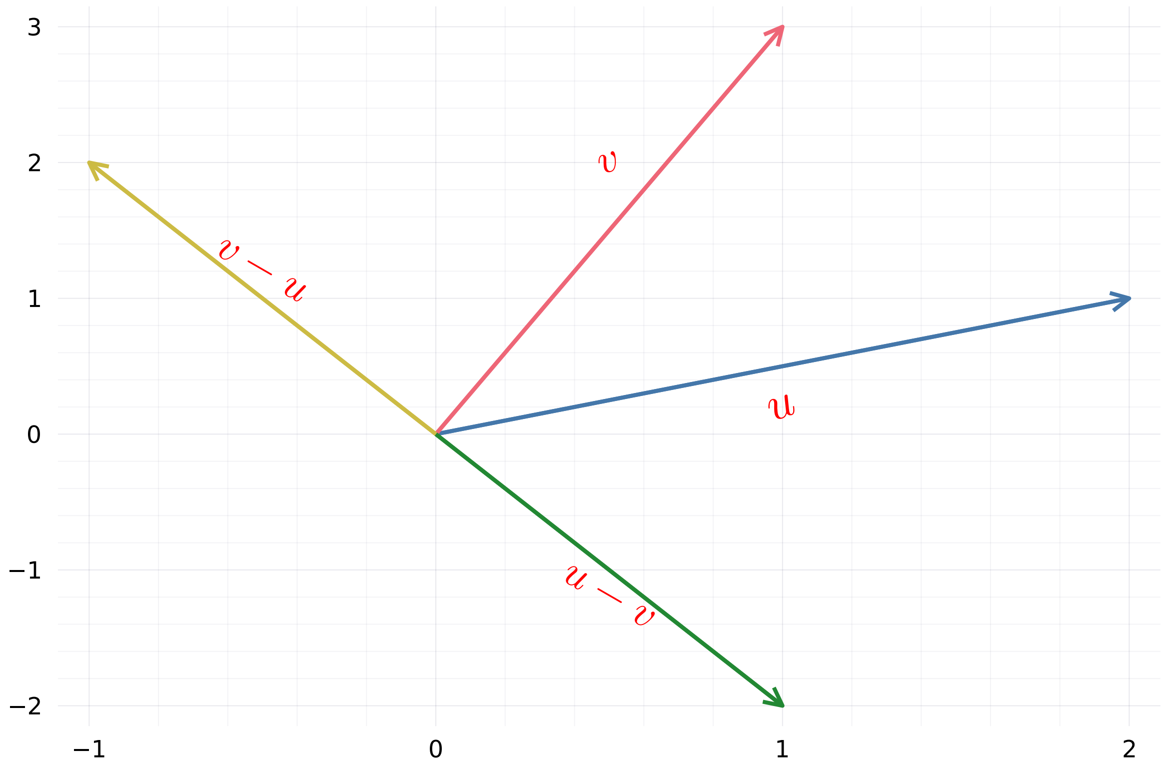



```
let
    zero=[0,0]
    u=[2,1]
    v=[1,3]
    sum1=u+v
    sum2=v+u
    ann=[
        (1,0.2,text(L"u",pointsize=16,rotation=20)),
        (0.5,2,text(L"v",pointsize=14,rotation=20)),
        (1.5,2.3,text(L"v+u",pointsize=14,rotation=45))
    ]
    p1=vec_plot(zero,u)
    p2=vec_plot!(zero,v)
    p3=vec_plot!(zero,sum2)
    p4=vec_plot!(u,sum2,:dash)
    p5=vec_plot!(v,sum2,:dash)
    p6=plot!(ann=ann,size=(320,320))

    save("vec_sum2",p6)

end
```

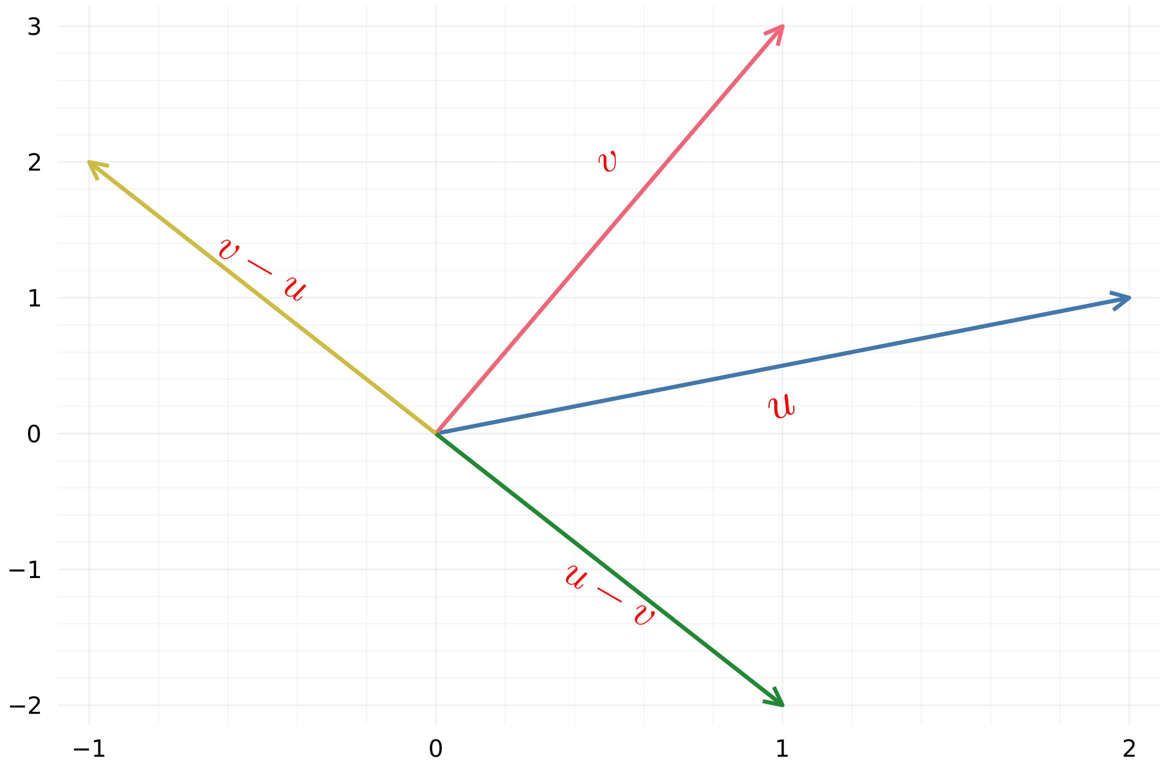
# 在坐标系中表示向量的差

两个向量的差与和不同, 顺序很重要,$u-v$ 与 $v-u$ 如果$v \neq u$, 则两者差指向相反的方向

如下图




```
md"""
## 在坐标系中表示向量的差

两个向量的差与和不同, 顺序很重要,$u-v$ 与 $v-u$  如果$v \neq u$, 则两者差指向相反的方向

如下图

$(store["vec_difference"])
"""
```




```julia
let
    zero,u,v=[0,0],[2,1],[1,3]
    diff1,diff2=u-v,v-u

    ann=[
        (1,0.2,text(L"u",pointsize=16,rotation=20,color=:red)),
        (0.5,2,text(L"v",pointsize=14,rotation=20,color=:red)),
        (0.5,-1.2,text(L"u-v",pointsize=14,rotation=-30,color=:red)),
        (-0.5,1.2,text(L"v-u",pointsize=14,rotation=-30,color=:red))
    ]
    p1=vec_plot(zero,u)
    p2=vec_plot!(zero,v)
    p3=vec_plot!(zero,diff1)
    #p4=vec_plot!(u,v,:dash)
    #p5=vec_plot!(u,diff1,:dash)
    p6=vec_plot!(zero,diff2)
    p7=plot!(ann=ann)
    save("vec_difference",p7)

end
```

# 向量加法的性质:

Definition

向量加法属性:

1. 加法交换律:

$$u + v = v + u$$

2. 加法结合律:

$$(u + v) + w = u + (v + w)$$

3. 存在0 向量使得:

$$u + 0 = u$$

4. 每个向量都有一个方向相反, 大小相同的向量使得:

$$u + (-u) = 0$$

```
vec_plot! (generic function with 2 methods)
```

```
begin
        store=Dict()

        function save(key::String, dict)
            store[key]=dict
        end

        function read(key::String)
            return  store[key]
        end


        function vec_plot(v1,v2,ls=:solid)
            v11,v12=v1[1],v1[2]
            v21,v22=v2[1],v2[2]
            return plot([v11,v21],[v12,v22],label=false, arrow=true, lw=2,ls=ls)
        end

        function vec_plot!(v1,v2,ls=:solid)
            v11,v12=v1[1],v1[2]
            v21,v22=v2[1],v2[2]
            return plot!([v11,v21],[v12,v22],label=false, arrow=true, lw=2,ls=ls)
        end




end
```