a ket vector is represented by n $\times$ 1 column matrice

$$|v\rangle = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_i \\ \vdots \\ v_n \end{pmatrix}.$$

a bra vector $\langle v|$

$$\begin{pmatrix} v_1^* & v_2^* & \dots & v_i^* & \dots & v_n^* \end{pmatrix}.$$

---

the inner product(内积) between two vectors:

$$c = \langle v|w\rangle = \sum_{i=1}^{n} v_i^* w_i,$$

code analysis

In/Out	Argument	Description
[in]	n	n (integer) is the dimension of the vectors between which we are calculating the inner product
[in]	v1	v1 is a real*8 array of dimension (0:n−1)
[in]	v2	v2 is a real*8 array of dimension (0:n−1)
[out]	c	c (real*8) is the inner product between vectors v1 and v2

```
subroutine innerproduct(v1,v2,n,c)
implicit none

integer :: n,i
real*8 :: v1(0:n-1),v2(0:n-1),c

c = 0.0d0

do i = 0,n-1
    c = c+ v1(i)*v2(i)
enddo

end subroutine innerproduct
```

Example of INNERPRODUCT

In this example we are finding the inner product between the vectors $v_1 = (1, 2, 3, 4)^T$ and $v_2 = (2, 3, 1, -1)^T$.

```
program example1
implicit none

real*8 :: v1(0:3),v2(0:3),c !v1,v2 = 1x4 column matrice

v1 = (/1.0, 2.0, 3.0, 4.0/)
v2 = (/2.0, 3.0, 1.0, -1.0/)

call innerproduct(v1,v2,4,c)

print*, c
! 7.0000000000000000

end program
```

In this example we are finding the inner product between the vectors $v_1 = (1, 1-i, 2+3i)^T$ and $v_2 = (2, 3-i, 2-3i)^T$.

```
subroutine innerproduct(v1,v2,n,c)
implicit none

integer :: n,i
```

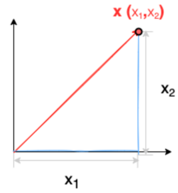
```fortran
complex*16 :: v1(0:n-1),v2(0:n-1),c

c = 0.0d0

do i = 0,n-1
    c = c+ dconjg(v1(i))*v2(i)
enddo

end subroutine innerproduct



program example2
implicit none

complex*16 :: v1(0:2),v2(0:2),c

v1(0) = dcmplx(1,0)
v1(1) = dcmplx(1,-1)
v1(2) = dcmplx(2,3)

v2(0) = dcmplx(2,0)
v2(1) = dcmplx(3,-1)
v2(2) = dcmplx(2,-3)

call innerproduct(v1,v2,3,c)

print*, c
! (1.0000000000000000,-10.000000000000000)

end program
```

---

the Euclidean norm of the vector: A **vector's length** is called the norm of the vector.

$$||v|| = \sqrt{\langle v|v\rangle} = \sqrt{\sum_{i=1}^{n} v_i^* v_i} = \sqrt{\sum_{i=1}^{n} |v_i|^2},$$

$$||x||_2 = \sqrt{x_1^2 + x_2^2}$$



向量范数

1. L_0 范数（也称0范数）

$$||x||_0 = \text{非零元素的个数}$$

2. L_1 范数（也称和范数或1范数）

$$||x||_1 = \sum_{i=1}^{m} |x_i| = |x_1| + \cdots + |x_m|$$

3. L_2 范数（常称 $Euclidean$ 范数，有时也称 $Frobenius$ 范数）

$$||x||_2 = (|x_1|^2 + \cdots + |x_m|^2)^{\frac{1}{2}}$$

code analysis

In/Out	Argument	Description
[in]	n	n (integer) is the dimension of the vector v
[in]	v	v is a real*8 array of dimension (0:n−1)
[out]	norm	norm (real*8) is the norm of the vector v

```
subroutine normvec(v,n,norm)
implicit none

integer :: n,i
real*8 :: v(0:n-1), c, norm

c = 0.0d0
norm = 0.0d0

do i = 0,n-1
    c = c+ v(i)*v(i)
enddo
norm = dsqrt(c)

end subroutine normvec
```

Example of NORM

In this example we are finding the norm of the vector $v = (1, 2, 3, 4)^T$.

```
program example1
implicit none

real*8 :: v(0:3),norm

v = (/1.0, 2.0, 3.0, 4.0/)

call normvec(v,4,norm)

print*, norm
! 5.4772255750516612

end program
```

In this example we are finding the norm of the vector $v = (1, 1 - i, 2 + 3i)^T$.

```
subroutine normvec(v,n,norm)
implicit none

integer :: n,i
complex*16 :: v(0:n-1), c
real*8 :: norm

c = 0.0d0
norm = 0.0d0

do i = 0,n-1
    c = c+ conjg(v(i))*v(i)
enddo
norm = dsqrt(abs(c))

end subroutine normvec



program example2
implicit none

```

```
complex*16 :: v(0:2)
real*8 :: norm

v(0) = dcmplx(1,0)
v(2) = dcmplx(2,3)
v(1) = dcmplx(1,-1)

call normvec(v,3,norm)

print*, norm
! 4.0000000000000000

end program
```

Once the norm is found, the normalized vector as:

$$|v'\rangle = \frac{|v\rangle}{||v||},$$

where $\langle v'|v'\rangle = 1$

code analysis

In/Out	Argument	Description
[in]	n	n (integer) is the dimension of the vector v
[in/out]	v	v is a real*8 array of dimension (0:n−1), on exit, it will be overwritten by the normalized vector

```
subroutine Normalization(v,n)
implicit none

integer :: n,i
real*8 :: v(0:n-1),c

c = 0.0d0

do i = 0:n-1
    c = c+ v(i)*v(i)
```

```
enddo
v = v/dsqrt(c)

end subroutine
```

Example of NORMALIZATION

In this example we are normalizing the vector $v = (1, 2, 3, 4)^T$.

```
program example1
implicit none

real*8 :: v(0:3)

v = (/1.0, 2.0, 3.0, 4.0/)

call normalization(v,4)
print*, v
!0.18257418583505536
!0.36514837167011072
!0.54772255750516607
!0.73029674334022143

end program
```

In this example we are normalizing the vector $v = (1, 1 - i, 2 + 3i)^T$.

```
subroutine Normalization(v,n)
implicit none

integer :: n,i
complex*16 :: v(0:n-1),c

c = 0.0d0

do i = 0,n-1
    c = c+ dconjg(v(i))*v(i)
enddo
v = v/dsqrt(abs(c))
```

```
end subroutine



program example2
implicit none

complex*16 :: v(0:2)

v(0) = dcmplx(1,0)
v(1) = dcmplx(1,-1)
v(2) = dcmplx(2,3)

call normalization(v,3)
print*, v
!(0.2500000000000000,0.0000000000000000)
!(0.2500000000000000,-0.2500000000000000)
!(0.5000000000000000,0.7500000000000000)

end program
```

---

outer product:

$$A = |v\rangle\langle w|,$$

A is a $n_1 \times n_2$ matrix (if n_1 is dimension of $|v\rangle$, n_2 is dimension of $\langle w|$)

Then, the ijth matrix element of A is given by:

$$A_{ij} = \langle i|A|j\rangle = v_i w_j^*.$$

you will find it many times in Quantum physics.

code analysis

In/Out	Argument	Description
[in]	n1	n1 (integer) is the dimension of the vector v
[in]	n2	n2 (integer) is the dimension of the vector w
[in]	v	v is a real*8 array of dimension (0:n1−1)
[in]	w	w is a real*8 array of dimension (0:n2−1)
[out]	A	A is a real*8 array of dimension (0:n1−1,0:n2−1), which is the outer product

```
subroutine outerproduct(v,w,n1,n2,A)
implicit none

integer :: n1,n2,i,j
real*8 :: v(0:n1-1),w(0:n2-1),A(0:n1-1,0:n2-1)

do i = 0,n1-1
    do j = 0,n2-1
        A(i,j) = v(i)*w(j)
    enddo
enddo
end subroutine
```

In this example we are finding the outer product between vectors $v_1 = (1, 2)^T$ and $v_2 = (2, 3, 1)^T$.

```
program example1
implicit none

real*8 :: v(0:1),w(0:2),A(0:1,0:2)

v = (/1.0, 2.0/)
w = (/2.0, 3.0, 1.0/)

call outerproduct(v,w,2,3,A)
print*, A
!2.0000000000000000 3.0000000000000000 1.0000000000000000
!4.0000000000000000 6.0000000000000000 2.0000000000000000

end program
```

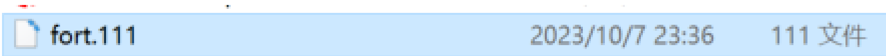
In this example we are finding the outer product between vectors $v_1 = (1, 1-i)^T$ and $v_2 = (2, 3-i)^T$.

```
subroutine outerproduct(v,w,n1,n2,A)
implicit none

integer :: n1,n2,i,j
complex*16 :: v(0:n1-1),w(0:n2-1),A(0:n1-1,0:n2-1)

A = 0.0d0
do i = 0,n1-1
    do j = 0,n2-1
        A(i,j) = v(i)*dconjg(w(j))
    enddo
enddo
end subroutine



program example1
implicit none

complex*16 :: v(0:1),w(0:2),A(0:1,0:1)

v(0) = dcmplx(1,0)
v(1) = dcmplx(1,-1)

w(0) = dcmplx(2,0)
w(1) = dcmplx(3,-1)

call outerproduct(v,w,2,2,A)
print*, A

end program
```

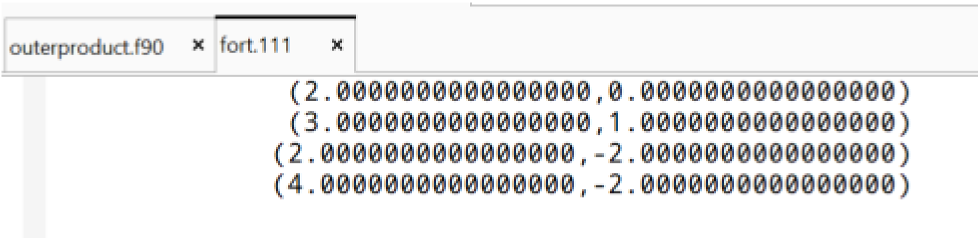
换一种print的方式

```
program example1
implicit none

complex*16 :: v(0:1),w(0:2),A(0:1,0:1)
integer :: i,j
!...
```

```fortran
call outerproduct(v,w,2,2,A)

do i = 0,1
    do j = 0,1
        write(111,*) A(i,j)
    enddo
enddo

end program
```



The file “`fort.111`” will contain the result:



these are elements of A matrix
