```
> restart;
> with(LinearAlgebra);
```

[*&x*, *Add*, *Adjoint*, *BackwardSubstitute*, *BandMatrix*, *Basis*, *BezoutMatrix*, *BidiagonalForm*, *BilinearForm*, *CARE*, *CharacteristicMatrix*, *CharacteristicPolynomial*, *Column*, *ColumnDimension*, *ColumnOperation*, *ColumnSpace*, *CompanionMatrix*, *ConditionNumber*, *ConstantMatrix*, *ConstantVector*, *Copy*, *CreatePermutation*, *CrossProduct*, *DARE*, *DeleteColumn*, *DeleteRow*, *Determinant*, *Diagonal*, *DiagonalMatrix*, *Dimension*, *Dimensions*, *DotProduct*, *EigenConditionNumbers*, *Eigenvalues*, *Eigenvectors*, *Equal*, *ForwardSubstitute*, *FrobeniusForm*, *GaussianElimination*, *GenerateEquations*, *GenerateMatrix*, *Generic*, *GetResultDataType*, *GetResultShape*, *GivensRotationMatrix*, *GramSchmidt*, *HankelMatrix*, *HermiteForm*, *HermitianTranspose*, *HessenbergForm*, *HilbertMatrix*, *HouseholderMatrix*, *IdentityMatrix*, *IntersectionBasis*, *IsDefinite*, *IsOrthogonal*, *IsSimilar*, *IsUnitary*, *JordanBlockMatrix*, *JordanForm*, *KroneckerProduct*, *LA_Main*, *LUDecomposition*, *LeastSquares*, *LinearSolve*, *LyapunovSolve*, *Map*, *Map2*, *MatrixAdd*, *MatrixExponential*, *MatrixFunction*, *MatrixInverse*, *MatrixMatrixMultiply*, *MatrixNorm*, *MatrixPower*, *MatrixScalarMultiply*, *MatrixVectorMultiply*, *MinimalPolynomial*, *Minor*, *Modular*, *Multiply*, *NoUserValue*, *Norm*, *Normalize*, *NullSpace*, *OuterProductMatrix*, *Permanent*, *Pivot*, *PopovForm*, *QRDecomposition*, *RandomMatrix*, *RandomVector*, *Rank*, *RationalCanonicalForm*, *ReducedRowEchelonForm*, *Row*, *RowDimension*, *RowOperation*, *RowSpace*, *ScalarMatrix*, *ScalarMultiply*, *ScalarVector*, *SchurForm*, *SingularValues*, *SmithForm*, *StronglyConnectedBlocks*, *SubMatrix*, *SubVector*, *SumBasis*, *SylvesterMatrix*, *SylvesterSolve*, *ToeplitzMatrix*, *Trace*, *Transpose*, *TridiagonalForm*, *UnitVector*, *VandermondeMatrix*, *VectorAdd*, *VectorAngle*, *VectorMatrixMultiply*, *VectorNorm*, *VectorScalarMultiply*, *ZeroMatrix*, *ZeroVector*, *Zip*] **(1)**

```
> Pauli1 := Matrix([[0,1],[1,0]]);Pauli2 := Matrix([[0,-I],[I,0]]);
  Pauli3 := Matrix([[1,0],[0,-1]]);Pauli0 := Matrix([[1,0],[0,1]]);
```

$$Pauli1 := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$Pauli2 := \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix}$$

$$Pauli3 := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$Pauli0 := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \textbf{(2)}$$

```
> Pauli2[1,2];
```

$$-I \quad \textbf{(3)}$$

```
> ProdInterno := sum (sum(''Pauli2''[i,k]*''Pauli2''[k,i],k=1..2),
  i=1..2);
```

$$ProdInterno := 2 \tag{4}$$

```
> Pauli1.Pauli1;
```

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{5}$$

```
> VectMas := Vector([1,0]);
```

$$VectMas := \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{6}$$

```
> VectMen := Vector([0,1]);
```

$$VectMen := \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{7}$$

```
> Pauli1.VectMas;Pauli1.VectMen;
```

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{8}$$

```
> Pauli2.VectMas;Pauli2.VectMen;
```

$$\begin{bmatrix} 0 \\ I \end{bmatrix}$$

$$\begin{bmatrix} -I \\ 0 \end{bmatrix} \tag{9}$$

```
> Pauli3.VectMas;Pauli3.VectMen;
```

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ -1 \end{bmatrix} \tag{10}$$

```
> M := Matrix([[M11,M12],[M21,M22]]);
```

$$M := \begin{bmatrix} M11 & M12 \\ M21 & M22 \end{bmatrix} \tag{11}$$

```
> CompM0 := sum (sum(''Pauli0''[i,k]*''M''[k,i],k=1..2),i=1..2);
```

$$CompM0 := M11 + M22 \tag{12}$$

```
> CompM1 := sum (sum(''Pauli1''[i,k]*''M''[k,i],k=1..2),i=1..2);
```

$$CompM1 := M21 + M12 \tag{13}$$

```
> CompM2 := sum (sum(''Pauli2''[i,k]*''M''[k,i],k=1..2),i=1..2);
```

$$CompM2 := -I\,M21 + I\,M12 \tag{14}$$

```
> CompM3 := sum (sum(''Pauli3''[i,k]*''M''[k,i],k=1..2),i=1..2);
```

$$CompM3 := M11 - M22 \tag{15}$$

```
> sum(1/(2*n)!,n=0..infinity);
```

$$\cosh(1) \tag{16}$$

```
> evalf(%);
```

$$1.543080635 \tag{17}$$

```
> 2*%;
```

$$3.086161270 \tag{18}$$

```
> sigmaP:= Pauli1 + I*Pauli2;
```

$$sigmaP := \begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix} \tag{19}$$

```
> sigmaM:= Pauli1 - I*Pauli2;;
```

$$sigmaM := \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix} \tag{20}$$

```
> PauliMas := Matrix([[Transpose(VectMas).sigmaP.VectMas, Transpose
(VectMas).sigmaP.VectMen],[Transpose(VectMen).sigmaP.VectMas,
Transpose(VectMen).sigmaP.VectMen]]);
```

$$PauliMas := \begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix} \tag{21}$$

```
> PauliMen := Matrix([[Transpose(VectMas).sigmaM.VectMas, Transpose
(VectMas).sigmaM.VectMen],[Transpose(VectMen).sigmaM.VectMas,
Transpose(VectMen).sigmaM.VectMen]]);
```

$$PauliMen := \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix} \tag{22}$$