#EX1:

```
> with(Student[LinearAlgebra]); with(LinearAlgebra)
[&x, `.`, AddRow, AddRows, Adjoint, ApplyLinearTransformPlot,
    BackwardSubstitute, BandMatrix, Basis, BilinearForm,
    CharacteristicMatrix, CharacteristicPolynomial, ColumnDimension,
    ColumnSpace, CompanionMatrix, ConstantMatrix, ConstantVector,
    CrossProduct, CrossProductPlot, Determinant, DeterminantSteps,
    Diagonal, DiagonalMatrix, Dimension, Dimensions, EigenPlot,
    EigenPlotTutor, Eigenvalues, EigenvaluesTutor, Eigenvectors,
    EigenvectorsTutor, Equal, GaussJordanEliminationTutor,
    GaussianElimination, GaussianEliminationTutor, GenerateEquations,
    GenerateMatrix, GramSchmidt, HermitianTranspose, HouseholderMatrix,
    Id, IdentityMatrix, IntersectionBasis, InverseTutor, IsDefinite,
    IsOrthogonal, IsSimilar, IsUnitary, JordanBlockMatrix, JordanForm,
    LUDecomposition, LeastSquares, LeastSquaresPlot, LinearSolve,
    LinearSolveTutor, LinearSystemPlot, LinearSystemPlotTutor,
    LinearTransformPlot, LinearTransformPlotTutor, MatrixBuilder,
    MatrixExponential, MatrixInverse, MinimalPolynomial, Minor,
    MultiplyRow, Norm, Normalize, NullSpace, Pivot, PlanePlot,
    ProjectionMatrix, ProjectionPlot, Pseudoinverse, QRDecomposition,
    RandomMatrix, RandomVector, Rank, ReducedRowEchelonForm,
    ReflectionMatrix, RotationMatrix, RowDimension, RowSpace, SetDefault,
    SetDefaults, SingularValues, SumBasis, SwapRow, SwapRows, Trace,
    Transpose, UnitVector, VectorAngle, VectorSumPlot, ZeroMatrix,
    ZeroVector]
[&x, Add, Adjoint, BackwardSubstitute, BandMatrix, Basis, BezoutMatrix,        (1)
    BidiagonalForm, BilinearForm, CARE, CharacteristicMatrix,
    CharacteristicPolynomial, Column, ColumnDimension, ColumnOperation,
    ColumnSpace, CompanionMatrix, CompressedSparseForm,
    ConditionNumber, ConstantMatrix, ConstantVector, Copy,
    CreatePermutation, CrossProduct, DARE, DeleteColumn, DeleteRow,
    Determinant, Diagonal, DiagonalMatrix, Dimension, Dimensions,
    DotProduct, EigenConditionNumbers, Eigenvalues, Eigenvectors, Equal,
    ForwardSubstitute, FrobeniusForm, FromCompressedSparseForm,
    FromSplitForm, 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,
ProjectionMatrix, QRDecomposition, RandomMatrix, RandomVector, Rank,
RationalCanonicalForm, ReducedRowEchelonForm, Row, RowDimension,
RowOperation, RowSpace, ScalarMatrix, ScalarMultiply, ScalarVector,
SchurForm, SingularValues, SmithForm, SplitForm,
StronglyConnectedBlocks, SubMatrix, SubVector, SumBasis,
SylvesterMatrix, SylvesterSolve, ToeplitzMatrix, Trace, Transpose,
TridiagonalForm, UnitVector, VandermondeMatrix, VectorAdd,
VectorAngle, VectorMatrixMultiply, VectorNorm, VectorScalarMultiply,
ZeroMatrix, ZeroVector, Zip]
```

```
> A := Matrix([[0,-2, 0], [1,-2, 0], [0, 0,-2]])
```

$$A := \begin{bmatrix} 0 & -2 & 0 \\ 1 & -2 & 0 \\ 0 & 0 & -2 \end{bmatrix} \qquad (2)$$

```
> eigenvals(Matrix(([[0,-2], [1,-2]])))
```

$$-1 + \mathrm{I}, -1 - \mathrm{I} \qquad (3)$$

```
> eigenvals(A);
```

$$-2, -1 - \mathrm{I}, -1 + \mathrm{I} \qquad (4)$$

#a)

```
> u1 := transpose(Matrix(vector([0, 0, 1])));
```

$$u1 := \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad (5)$$

```
> lambda[1] :=-2;
```

$$\lambda_1 := -2 \qquad (6)$$

```
> evalm(A &* u1 − lambda1·u1);
```

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{7}$$

```
> #b)
```

```
> u2 := transpose(Matrix([1 + i, 1, 0]));
```

$$u2 := \begin{bmatrix} 1+i \\ 1 \\ 0 \end{bmatrix} \tag{8}$$

```
> lambda[2] := −1 + i;
```

$$\lambda_2 := -1+i \tag{9}$$

```
> evalm(A &* u2 − lambda[2]·u2);
```

$$\begin{bmatrix} -2+(1-i)\,(1+i) \\ 0 \\ 0 \end{bmatrix} \tag{10}$$

```
> simplify(%);
      # sau expand(%)% inseamna sa simplificam ultima comanda executata
```

$$\begin{bmatrix} -i^2-1 \\ 0 \\ 0 \end{bmatrix} \tag{11}$$

```
> #c)
> #d)
> lambda[3] := −1 − i
```

$$\lambda_3 := -1-i \tag{12}$$

```
> u3 := transpose(Matrix([1 − i, 1, 0]));
```

$$u3 := \begin{bmatrix} 1-i \\ 1 \\ 0 \end{bmatrix} \tag{13}$$

```
> P := Matrix([u1, u2, u3]);
```

$$P := \begin{bmatrix} 0 & 1+i & 1-i \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \tag{14}$$

```
> #e)
> J := DiagonalMatrix([lambda[1], lambda[2], lambda[3]]);
```

$$J := \begin{bmatrix} -2 & 0 & 0 \\ 0 & -1+i & 0 \\ 0 & 0 & -1-i \end{bmatrix} \tag{15}$$

```
> J;
```

$$\begin{bmatrix} -2 & 0 & 0 \\ 0 & -1+i & 0 \\ 0 & 0 & -1-i \end{bmatrix} \tag{16}$$

```
> eigenvals(J);
```

$$-2,\ -1+i,\ -1-i \tag{17}$$

```
> #f)-----------------------------------------------------------------
> A1 := evalm(P &* J );
```

$$A1 := \begin{bmatrix} 0 & (1+i)\,(-1+i) & (1-i)\,(-1-i) \\ 0 & -1+i & -1-i \\ -2 & 0 & 0 \end{bmatrix} \tag{18}$$

```
> evalm(A1 &* P^(-1))
```

$$\begin{bmatrix} \frac{(1+i)\,(-1+i)}{2\,i} - \frac{(1-i)\,(-1-i)}{2\,i} & \frac{(1+i)\,(-1+i)}{2\,i} \dots \\ \frac{-1+i}{2\,i} - \frac{-1-i}{2\,i} & \frac{(-1+i)}{2\,i} \dots \\ 0 & \dots \end{bmatrix} \tag{19}$$

```
> B := simplify(%);
```

$$B := \begin{bmatrix} 0 & i^2 - 1 & 0 \\ 1 & -2 & 0 \\ 0 & 0 & -2 \end{bmatrix} \tag{20}$$

```
> evalm(A - B);
```

$$\begin{bmatrix} 0 & -i^2 - 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{21}$$

```
>
> #g)-----------------------------------------------------------
> MatrixExponential(t·J);
```

$$\begin{bmatrix} e^{-2\,t} & 0 & 0 \\ 0 & e^{t\,(-1+i)} & 0 \\ 0 & 0 & e^{t\,(-1-i)} \end{bmatrix} \tag{22}$$

```
> exp(t·J)
```

$$\begin{bmatrix} e^{-2\,t} & 1 & 1 \\ 1 & e^{t\,(-1+i)} & 1 \\ 1 & 1 & e^{t\,(-1-i)} \end{bmatrix} \tag{23}$$

```
> t·J;
```

$$\begin{bmatrix} -2\,t & 0 & 0 \\ 0 & t\,(-1+i) & 0 \\ 0 & 0 & t\,(-1-i) \end{bmatrix} \tag{24}$$

```
> t·A;
```

(25)

$$\begin{bmatrix} 0 & -2\,t & 0 \\ t & -2\,t & 0 \\ 0 & 0 & -2\,t \end{bmatrix} \qquad (25)$$

```
> exp(t·A);
```

$$\begin{bmatrix} 1 & e^{-2\,t} & 1 \\ e^{t} & e^{-2\,t} & 1 \\ 1 & 1 & e^{-2\,t} \end{bmatrix} \qquad (26)$$

```
> C := MatrixExponential(t·A);
     #sunt valori complexe deoarece eigenvalues is complexe
```

$$C := \begin{bmatrix} e^{-t}\cos(t) + e^{-t}\sin(t) & -2\,e^{-t}\sin(t) & 0 \\ e^{-t}\sin(t) & e^{-t}\cos(t) - e^{-t}\sin(t) & 0 \\ 0 & 0 & e^{-2\,t} \end{bmatrix} \qquad (27)$$

```
> #h)
> for i from 1 to 3 do
      for j from 1 to 3 do
         lim := limit(C[i][j], t = infinity);
         print(lim);
       end do;
   end do;
Error, (in limit) invalid limiting point
>
>
> #i)
> ec1 := diff(x[i](t), t) =−2·x[2](t)
```

$$ec1 := \frac{d}{dt}\, x_1(t) = -2\, x_2(t) \qquad (28)$$

```
> ec2 := diff(x[2](t), t) = x[1](t) − 2·x[2](t)
```

$$ec2 := \frac{d}{dt}\, x_2(t) = x_1(t) - 2\, x_2(t) \qquad (29)$$

```
> ec3 := diff(x[3](t), t) = −2·x[3](t);
```

$$ec3 := \frac{d}{dt}\, x_3(t) = -2\, x_3(t) \qquad (30)$$

```
> sist := ec1, ec2, ec3
```

$$sist := \frac{\mathrm{d}}{\mathrm{d}t}\, x_1(t) = -2\, x_2(t),\ \frac{\mathrm{d}}{\mathrm{d}t}\, x_2(t) = x_1(t) - 2\, x_2(t),\ \frac{\mathrm{d}}{\mathrm{d}t}\, x_3(t) = -2\, x_3(t) \qquad \textbf{(31)}$$

```
> dsolve({sist}, {x[1], x[2], x[3]})
Error, (in StringTools:-IsPrefix) second argument must be a string
>
>
>
>
>
> #PB2 ------------------------------------------------------------
> with(Student[LinearAlgebra]); with(LinearAlgebra)
```

```
[&x, `.`, AddRow, AddRows, Adjoint, ApplyLinearTransformPlot,
    BackwardSubstitute, BandMatrix, Basis, BilinearForm,
    CharacteristicMatrix, CharacteristicPolynomial, ColumnDimension,
    ColumnSpace, CompanionMatrix, ConstantMatrix, ConstantVector,
    CrossProduct, CrossProductPlot, Determinant, DeterminantSteps,
    Diagonal, DiagonalMatrix, Dimension, Dimensions, EigenPlot,
    EigenPlotTutor, Eigenvalues, EigenvaluesTutor, Eigenvectors,
    EigenvectorsTutor, Equal, GaussJordanEliminationTutor,
    GaussianElimination, GaussianEliminationTutor, GenerateEquations,
    GenerateMatrix, GramSchmidt, HermitianTranspose,
    HouseholderMatrix, Id, IdentityMatrix, IntersectionBasis, InverseTutor,
    IsDefinite, IsOrthogonal, IsSimilar, IsUnitary, JordanBlockMatrix,
    JordanForm, LUDecomposition, LeastSquares, LeastSquaresPlot,
    LinearSolve, LinearSolveTutor, LinearSystemPlot,
    LinearSystemPlotTutor, LinearTransformPlot,
    LinearTransformPlotTutor, MatrixBuilder, MatrixExponential,
    MatrixInverse, MinimalPolynomial, Minor, MultiplyRow, Norm,
    Normalize, NullSpace, Pivot, PlanePlot, ProjectionMatrix, ProjectionPlot,
    Pseudoinverse, QRDecomposition, RandomMatrix, RandomVector, Rank,
    ReducedRowEchelonForm, ReflectionMatrix, RotationMatrix,
    RowDimension, RowSpace, SetDefault, SetDefaults, SingularValues,
    SumBasis, SwapRow, SwapRows, Trace, Transpose, UnitVector,
    VectorAngle, VectorSumPlot, ZeroMatrix, ZeroVector]
[&x, Add, Adjoint, BackwardSubstitute, BandMatrix, Basis, BezoutMatrix,        (32)
    BidiagonalForm, BilinearForm, CARE, CharacteristicMatrix,
    CharacteristicPolynomial, Column, ColumnDimension, ColumnOperation,
    ColumnSpace, CompanionMatrix, CompressedSparseForm,
    ConditionNumber, ConstantMatrix, ConstantVector, Copy,
    CreatePermutation, CrossProduct, DARE, DeleteColumn, DeleteRow,
    Determinant, Diagonal, DiagonalMatrix, Dimension, Dimensions,
    DotProduct, EigenConditionNumbers, Eigenvalues, Eigenvectors, Equal,
    ForwardSubstitute, FrobeniusForm, FromCompressedSparseForm,
    FromSplitForm, 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, ProjectionMatrix, QRDecomposition, RandomMatrix,
    RandomVector, Rank, RationalCanonicalForm,
    ReducedRowEchelonForm, Row, RowDimension, RowOperation,
    RowSpace, ScalarMatrix, ScalarMultiply, ScalarVector, SchurForm,
    SingularValues, SmithForm, SplitForm, StronglyConnectedBlocks,
    SubMatrix, SubVector, SumBasis, SylvesterMatrix, SylvesterSolve,
    ToeplitzMatrix, Trace, Transpose, TridiagonalForm, UnitVector,
    VandermondeMatrix, VectorAdd, VectorAngle, VectorMatrixMultiply,
    VectorNorm, VectorScalarMultiply, ZeroMatrix, ZeroVector, Zip]
> restart;
> with(Student[LinearAlgebra]); with(LinearAlgebra)
[&x, `.`, AddRow, AddRows, Adjoint, ApplyLinearTransformPlot,
    BackwardSubstitute, BandMatrix, Basis, BilinearForm,
    CharacteristicMatrix, CharacteristicPolynomial, ColumnDimension,
    ColumnSpace, CompanionMatrix, ConstantMatrix, ConstantVector,
    CrossProduct, CrossProductPlot, Determinant, DeterminantSteps,
    Diagonal, DiagonalMatrix, Dimension, Dimensions, EigenPlot,
    EigenPlotTutor, Eigenvalues, EigenvaluesTutor, Eigenvectors,
    EigenvectorsTutor, Equal, GaussJordanEliminationTutor,
    GaussianElimination, GaussianEliminationTutor, GenerateEquations,
    GenerateMatrix, GramSchmidt, HermitianTranspose,
    HouseholderMatrix, Id, IdentityMatrix, IntersectionBasis, InverseTutor,
    IsDefinite, IsOrthogonal, IsSimilar, IsUnitary, JordanBlockMatrix,
    JordanForm, LUDecomposition, LeastSquares, LeastSquaresPlot,
    LinearSolve, LinearSolveTutor, LinearSystemPlot,
    LinearSystemPlotTutor, LinearTransformPlot,
    LinearTransformPlotTutor, MatrixBuilder, MatrixExponential,
    MatrixInverse, MinimalPolynomial, Minor, MultiplyRow, Norm,
    Normalize, NullSpace, Pivot, PlanePlot, ProjectionMatrix, ProjectionPlot,
    Pseudoinverse, QRDecomposition, RandomMatrix, RandomVector, Rank,
    ReducedRowEchelonForm, ReflectionMatrix, RotationMatrix,
    RowDimension, RowSpace, SetDefault, SetDefaults, SingularValues,
    SumBasis, SwapRow, SwapRows, Trace, Transpose, UnitVector,
    VectorAngle, VectorSumPlot, ZeroMatrix, ZeroVector]
[&x, Add, Adjoint, BackwardSubstitute, BandMatrix, Basis, BezoutMatrix,      (33)
    BidiagonalForm, BilinearForm, CARE, CharacteristicMatrix,
    CharacteristicPolynomial, Column, ColumnDimension, ColumnOperation,
    ColumnSpace, CompanionMatrix, CompressedSparseForm,
    ConditionNumber, ConstantMatrix, ConstantVector, Copy,
    CreatePermutation, CrossProduct, DARE, DeleteColumn, DeleteRow,
    Determinant, Diagonal, DiagonalMatrix, Dimension, Dimensions,
    DotProduct, EigenConditionNumbers, Eigenvalues, Eigenvectors, Equal,
    ForwardSubstitute, FrobeniusForm, FromCompressedSparseForm,
    FromSplitForm, 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, ProjectionMatrix, QRDecomposition, RandomMatrix,
    RandomVector, Rank, RationalCanonicalForm,
    ReducedRowEchelonForm, Row, RowDimension, RowOperation,
    RowSpace, ScalarMatrix, ScalarMultiply, ScalarVector, SchurForm,
    SingularValues, SmithForm, SplitForm, StronglyConnectedBlocks,
```

SubMatrix, SubVector, SumBasis, SylvesterMatrix, SylvesterSolve, ToeplitzMatrix, Trace, Transpose, TridiagonalForm, UnitVector, VandermondeMatrix, VectorAdd, VectorAngle, VectorMatrixMultiply, VectorNorm, VectorScalarMultiply, ZeroMatrix, ZeroVector, Zip]

```
> D1 := DiagonalMatrix([2, 2,−1, 0]);
```

$$D1 := \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{34}$$

```
> P := RandomMatrix(4, generator = −5 ..5);
```

$$P := \begin{bmatrix} -2 & -4 & 1 & 2 \\ 4 & -3 & 4 & 3 \\ 3 & -1 & 0 & -2 \\ -2 & -2 & -1 & -1 \end{bmatrix} \tag{35}$$

```
> while Determinant(P) = 0 do
    P := RandomMatrix(4, generator = −5 ..5)
  end do;
>
```

```
> A := simplify(P . D1 . P^(−1))
```

$$A := \begin{bmatrix} \frac{61}{19} & -\frac{26}{19} & \frac{34}{19} & -\frac{24}{19} \\ \frac{462}{19} & -\frac{286}{19} & \frac{336}{19} & -\frac{606}{19} \\ \frac{148}{19} & -\frac{88}{19} & \frac{118}{19} & -\frac{204}{19} \\ -\frac{97}{19} & \frac{70}{19} & -\frac{74}{19} & \frac{164}{19} \end{bmatrix} \tag{36}$$

```
> eigenvals(A);
```

$$\mathit{eigenvals}\left(\begin{bmatrix} \frac{61}{19} & -\frac{26}{19} & \frac{34}{19} & -\frac{24}{19} \\ \frac{462}{19} & -\frac{286}{19} & \frac{336}{19} & -\frac{606}{19} \\ \frac{148}{19} & -\frac{88}{19} & \frac{118}{19} & -\frac{204}{19} \\ -\frac{97}{19} & \frac{70}{19} & -\frac{74}{19} & \frac{164}{19} \end{bmatrix}\right) \tag{37}$$

```
> IsDiagonal(A)
```

$$\mathit{IsDiagonal}\left(\begin{bmatrix} \frac{61}{19} & -\frac{26}{19} & \frac{34}{19} & -\frac{24}{19} \\ \frac{462}{19} & -\frac{286}{19} & \frac{336}{19} & -\frac{606}{19} \\ \frac{148}{19} & -\frac{88}{19} & \frac{118}{19} & -\frac{204}{19} \\ -\frac{97}{19} & \frac{70}{19} & -\frac{74}{19} & \frac{164}{19} \end{bmatrix}\right) \tag{38}$$

```
>
```

```
> Determinant(A);
```

$$0 \tag{39}$$

```
> CharacteristicPolynomial(A, lambda);
```

$$\lambda^4 - 3\lambda^3 + 4\lambda \tag{40}$$

```
> eigenvectors(A);
```

$$\mathit{eigenvectors}\left(\begin{bmatrix} \frac{61}{19} & -\frac{26}{19} & \frac{34}{19} & -\frac{24}{19} \\ \frac{462}{19} & -\frac{286}{19} & \frac{336}{19} & -\frac{606}{19} \\ \frac{148}{19} & -\frac{88}{19} & \frac{118}{19} & -\frac{204}{19} \\ -\frac{97}{19} & \frac{70}{19} & -\frac{74}{19} & \frac{164}{19} \end{bmatrix}\right) \tag{41}$$

```
> JordanForm(A);
```

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \tag{42}$$

```
> evalm(A^-1);
Error, (in rtable/Power) singular matrix
> #pt ca nu i inversabila
>
>
>
>
> #de rez 3, 4 si ce a mai ramans
>
> #EX3:
> restart;
> with(LinearAlgebra)
[&x, Add, Adjoint, BackwardSubstitute, BandMatrix, Basis, BezoutMatrix,      (43)
    BidiagonalForm, BilinearForm, CARE, CharacteristicMatrix,
    CharacteristicPolynomial, Column, ColumnDimension, ColumnOperation,
    ColumnSpace, CompanionMatrix, CompressedSparseForm,
    ConditionNumber, ConstantMatrix, ConstantVector, Copy,
    CreatePermutation, CrossProduct, DARE, DeleteColumn, DeleteRow,
    Determinant, Diagonal, DiagonalMatrix, Dimension, Dimensions,
    DotProduct, EigenConditionNumbers, Eigenvalues, Eigenvectors, Equal,
    ForwardSubstitute, FrobeniusForm, FromCompressedSparseForm,
    FromSplitForm, 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*, *ProjectionMatrix*, *QRDecomposition*, *RandomMatrix*, *RandomVector*, *Rank*, *RationalCanonicalForm*, *ReducedRowEchelonForm*, *Row*, *RowDimension*, *RowOperation*, *RowSpace*, *ScalarMatrix*, *ScalarMultiply*, *ScalarVector*, *SchurForm*, *SingularValues*, *SmithForm*, *SplitForm*, *StronglyConnectedBlocks*, *SubMatrix*, *SubVector*, *SumBasis*, *SylvesterMatrix*, *SylvesterSolve*, *ToeplitzMatrix*, *Trace*, *Transpose*, *TridiagonalForm*, *UnitVector*, *VandermondeMatrix*, *VectorAdd*, *VectorAngle*, *VectorMatrixMultiply*, *VectorNorm*, *VectorScalarMultiply*, *ZeroMatrix*, *ZeroVector*, *Zip*]

```
> #a)
> A := Matrix([[2, 1, 0], [0, 2, 1], [0, 0, 2]]);
```

$$A := \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{bmatrix} \tag{44}$$

```
>
```

```
> #b)
> B := Matrix([[0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1], [-1, 0,-2, 0]]);
```

$$B := \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & -2 & 0 \end{bmatrix} \tag{45}$$

```
> EV := Eigenvectors(B)
```

$$EV := \begin{bmatrix} \mathrm{I} \\ \mathrm{I} \\ -\mathrm{I} \\ -\mathrm{I} \end{bmatrix}, \begin{bmatrix} \mathrm{I} & 0 & -\mathrm{I} & 0 \\ -1 & 0 & -1 & 0 \\ -\mathrm{I} & 0 & \mathrm{I} & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \tag{46}$$

```
> EV[1];#valorile proprii(lambdas)
```

(47)

$$\begin{bmatrix} I \\ I \\ -I \\ -I \end{bmatrix} \tag{47}$$

> *EV*[2];*#vectorii proprii(coloane)*

$$\begin{bmatrix} I & 0 & -I & 0 \\ -1 & 0 & -1 & 0 \\ -I & 0 & I & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \tag{48}$$

> $J := JordanForm(B);$

$$J := \begin{bmatrix} -I & 1 & 0 & 0 \\ 0 & -I & 0 & 0 \\ 0 & 0 & I & 1 \\ 0 & 0 & 0 & I \end{bmatrix} \tag{49}$$

> $\exp(tA) := MatrixExponential(t \cdot A);$

$$\mathrm{e}^{tA} := \begin{bmatrix} \mathrm{e}^{2t} & t\,\mathrm{e}^{2t} & \frac{t^2\,\mathrm{e}^{2t}}{2} \\ 0 & \mathrm{e}^{2t} & t\,\mathrm{e}^{2t} \\ 0 & 0 & \mathrm{e}^{2t} \end{bmatrix} \tag{50}$$

>

> $evalf(\exp(tA));$

$$\begin{bmatrix} \mathrm{e}^{2.t} & t\,\mathrm{e}^{2.t} & 0.5000000000\,t^2\,\mathrm{e}^{2.t} \\ 0. & \mathrm{e}^{2.t} & t\,\mathrm{e}^{2.t} \\ 0. & 0. & \mathrm{e}^{2.t} \end{bmatrix} \tag{51}$$

>

>

>

```
> 
> 
> #------------------------------------------------------------EX4:-
> 
> restart;
> with(DEtools) :
> 
> #======i)
> #definirea ecuatiei diferentiale
> eq := diff(x(t), t) = 1 - x(t)^2;
```

$$eq := \frac{\mathrm{d}}{\mathrm{d}t}\, x(t) = 1 - x(t)^2 \tag{52}$$

```
>  #rezolvarea generala
> sol := dsolve(eq, x(t));
```

$$sol := x(t) = \tanh(t + c_1) \tag{53}$$

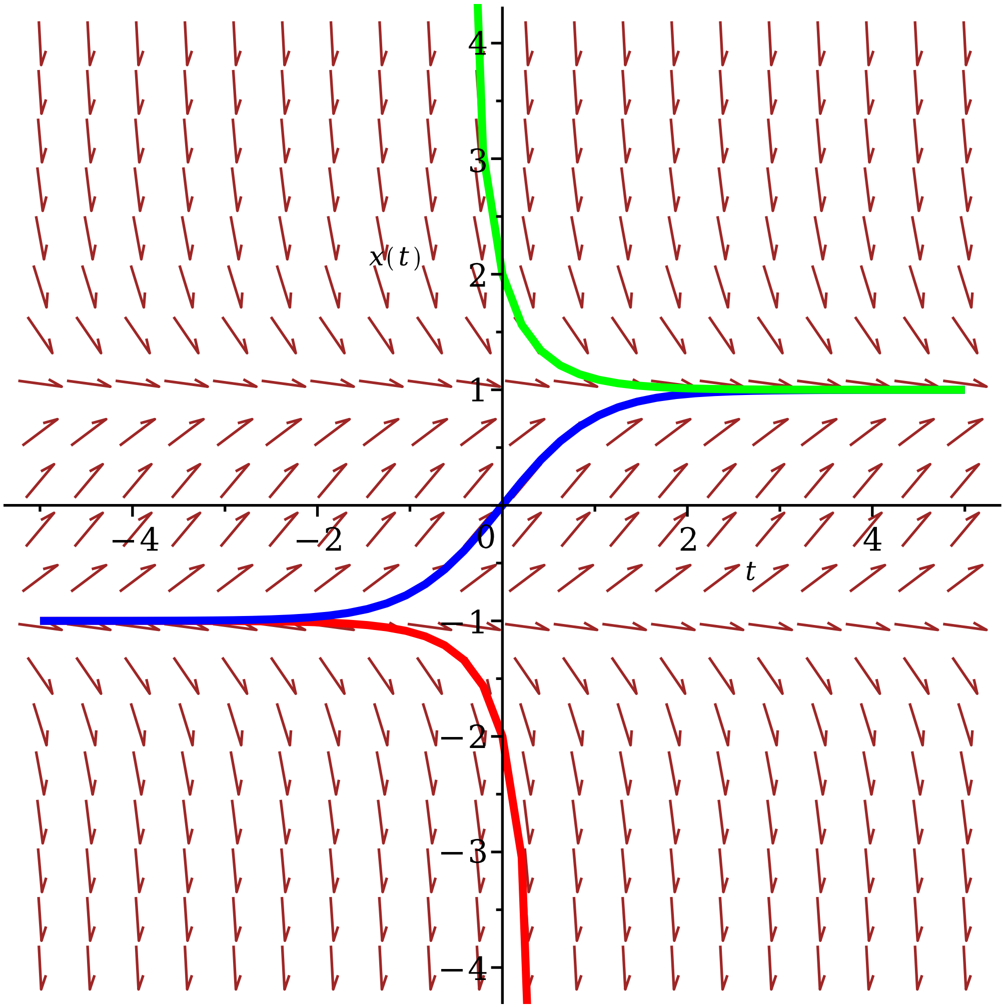
```
> 
> 
> #======ii)
> 
>  #φ(t,−2);
> sol_minus2 := dsolve( {diff(x(t), t) = 1 − x(t)^2, x(0 ) =−2}, x(t) ) :
> phi_minus2 := rhs(sol_minus2) :
> 
>  #φ(t, 0);
> sol_0 := dsolve( {diff(x(t), t) = 1 − x(t)^2, x(0 ) = 0}, x(t) ) :
> phi_0 := rhs(sol_0) :
> 
> 
>  #φ(t, 2) :
> sol2 := dsolve( {diff(x(t), t) = 1 − x(t)^2, x(0 ) =−2}, x(t) ) :
> phi2 := rhs(sol2) :
> 
> 
> 
> #======iii)

with(DEtools);
```

[AreSimilar, Closure, DEnormal, DEplot, DEplot3d, DEplot_polygon, DFactor, DFactorLCLM, DFactorsols, Dchangevar, Desingularize, FindODE, FunctionDecomposition, GCRD, Gosper, Heunsols, Homomorphisms, IVPsol, IsHyperexponential, LCLM, MeijerGsols, MultiplicativeDecomposition, ODEInvariants, PDEchangecoords, **(54)**

```
PolynomialNormalForm, RationalCanonicalForm, ReduceHyperexp,
RiemannPsols, Xchange, Xcommutator, Xgauge, Zeilberger, abelsol,
adjoint, autonomous, bernoullisol, buildsol, buildsym, canoni, caseplot,
casesplit, checkrank, chinisol, clairautsol, constcoeffsols, convertAlg,
convertsys, dalembertsol, dcoeffs, de2diffop, dfieldplot, diff_table,
diffop2de, dperiodic_sols, dpolyform, dsubs, eigenring,
endomorphism_charpoly, equinv, eta_k, eulersols, exactsol, expsols,
exterior_power, firint, firtest, formal_sol, gen_exp, generate_ic,
genhomosol, gensys, hamilton_eqs, hypergeometricsols, hypergeomsols,
hyperode, indicialeq, infgen, initialdata, integrate_sols, intfactor,
invariants, kovacicsols, leftdivision, liesol, line_int, linearsol, matrixDE,
matrix_riccati, maxdimsystems, moser_reduce, muchange, mult, mutest,
newton_polygon, normalG2, ode_int_y, ode_y1, odeadvisor, odepde,
parametricsol, particularsol, phaseportrait, poincare, polysols,
power_equivalent, rational_equivalent, ratsols, redode, reduceOrder,
reduce_order, regular_parts, regularsp, remove_RootOf, riccati_system,
riccatisol, rifread, rifsimp, rightdivision, rtaylor, separablesol,
singularities, solve_group, super_reduce, symgen, symmetric_power,
symmetric_product, symtest, transinv, translate, untranslate, varparam,
zoom]
```

```
> DEplot(
      diff(x(t), t) = 1 − x(t)^2,
      x(t),
      t = −5 ..5,
      [[x(0) = −2], [x(0) = 0], [x(0) = 2]],
      x = −4 ..4,
      linecolor = [red, blue, green]
  );
```



```
> 
> 
> #======iv)
> 
> limit(phi_minus2, t = −infinity);
                                 −1                                      (55)
> limit(phi2, t = infinity);
                                  1                                      (56)
> limit(phi_0, t = −infinity);
                                 −1                                      (57)
> limit(phi_0, t = infinity);
                                  1                                      (58)
> 
> 
> #======v)
> diff(phi_0, t);
                             1 − tanh(t)^2                               (59)
> diff(phi2, t);
                                                                         (60)
```

$$1-\coth\left(-\operatorname{arctanh}\left(\frac{1}{2}\right)+t\right)^{2} \tag{60}$$

```
> diff(phi_minus2, t);
```

$$1-\coth\left(-\operatorname{arctanh}\left(\frac{1}{2}\right)+t\right)^{2} \tag{61}$$

```
> limit(phi_0, t = −infinity);
```

$$-1 \tag{62}$$

```
> limit(phi_0, t = infinity);
```

$$1 \tag{63}$$

```
> limit(phi2, t = −infinity);
```

$$-1 \tag{64}$$

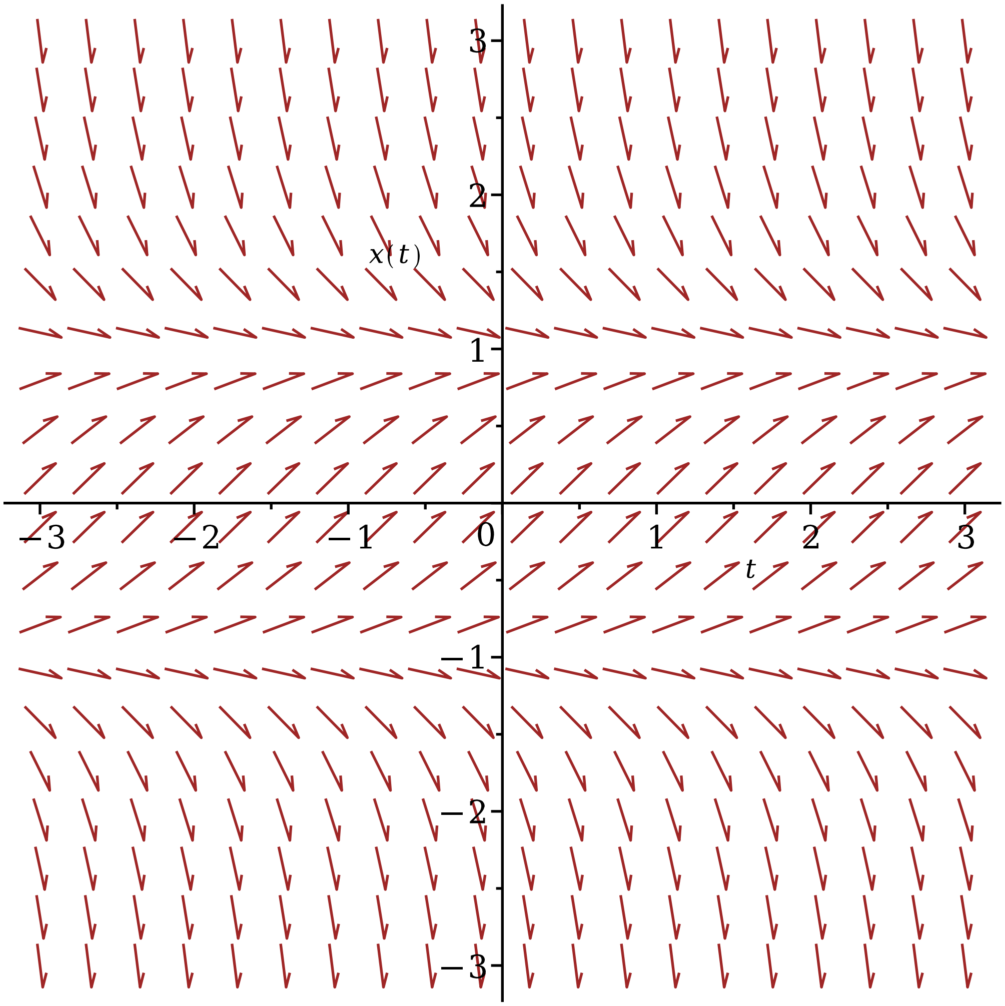
```
> limit(phi_minus2, t = −infinity);
```

$$-1 \tag{65}$$

```
> limit(phi_minus2, t = infinity);
```

$$1 \tag{66}$$

```
>
>
> #======vi)
> with(DEtools) :
> dfieldplot(diff(x(t), t) = 1 − x(t)^2, x(t), t = −3 ..3, x = −3 ..3);
```



3
2
x(t)
1
−3
−2
−1
0
1
2
3
t
−1
−2
−3

>

>