```
>
> #EX1:
> #a)
> restart;
> eq1 := 2·x − x^2 − x·y
> eq2 := −y + x·y
> solve({2·x − x^2 − x·y = 0, −y + x·y = 0}, {x, y});
```

$$\{x = 0, y = 0\}, \{x = 2, y = 0\}, \{x = 1, y = 1\} \quad (1)$$

```
>
> #b)
> 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, 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] (2)

```
> with(VectorCalculus)
```

[&x, `*`, `+`, `-`, `.`, <,>, <|>, About, AddCoordinates, ArcLength, (3)

BasisFormat, Binormal, ConvertVector, CrossProduct, Curl, Curvature, D, *Del, DirectionalDiff, Divergence, DotProduct, Flux, GetCoordinateParameters, GetCoordinates, GetNames, GetPVDescription, GetRootPoint, GetSpace, Gradient, Hessian, IsPositionVector, IsRootedVector, IsVectorField, Jacobian, Laplacian, LineInt, MapToBasis,*∇*, Norm, Normalize, PathInt, PlotPositionVector, PlotVector, PositionVector, PrincipalNormal, RadiusOfCurvature, RootedVector, ScalarPotential, SetCoordinateParameters, SetCoordinates, SpaceCurve, SurfaceInt, TNBFrame, TangentLine, TangentPlane, TangentVector, Torsion, Vector, VectorField, VectorPotential, VectorSpace, Wronskian, diff, eval, evalVF, int, limit, series*]

```
> with(linalg)
```

[*BlockDiagonal, GramSchmidt, JordanBlock, LUdecomp, QRdecomp, Wronskian, addcol, addrow, adj, adjoint, angle, augment, backsub, band, basis, bezout, blockmatrix, charmat, charpoly, cholesky, col, coldim, colspace, colspan, companion, concat, cond, copyinto, crossprod, curl, definite, delcols, delrows, det, diag, diverge, dotprod, eigenvals, eigenvalues, eigenvectors, eigenvects, entermatrix, equal, exponential, extend, ffgausselim, fibonacci, forwardsub, frobenius, gausselim, gaussjord, geneqns, genmatrix, grad, hadamard, hermite, hessian, hilbert, htranspose, ihermite, indexfunc, innerprod, intbasis, inverse, ismith, issimilar, iszero, jacobian, jordan, kernel, laplacian, leastsqrs, linsolve, matadd, matrix, minor, minpoly, mulcol, mulrow, multiply, norm, normalize, nullspace, orthog, permanent, pivot, potential, randmatrix, randvector, rank, ratform, row, rowdim, rowspace, rowspan, rref, scalarmul, singularvals, smith, stackmatrix, submatrix, subvector, sumbasis, swapcol, swaprow, sylvester, toeplitz, trace, transpose, vandermonde, vecpotent, vectdim, vector, wronskian*] **(4)**

```
> Jm := Jacobian([2·x − x^2 − x·y, −y + x·y], [x, y]);
```

$$Jm := \begin{bmatrix} -2x - y + 2 & -x \\ y & x - 1 \end{bmatrix} \quad \textbf{(5)}$$

```
>
> J1 := subs([x = 0, y = 0], Jm);
```

$$J1 := \begin{bmatrix} 2 & 0 \\ 0 & -1 \end{bmatrix} \quad \textbf{(6)}$$

```
> eigenvalues(J1);
```

$$2, -1 \tag{7}$$

```
> J2 := subs([x = 2, y = 0], Jm)
```

$$J2 := \begin{bmatrix} -2 & -2 \\ 0 & 1 \end{bmatrix} \tag{8}$$

```
> eigenvalues(J2);
```

$$-2, 1 \tag{9}$$

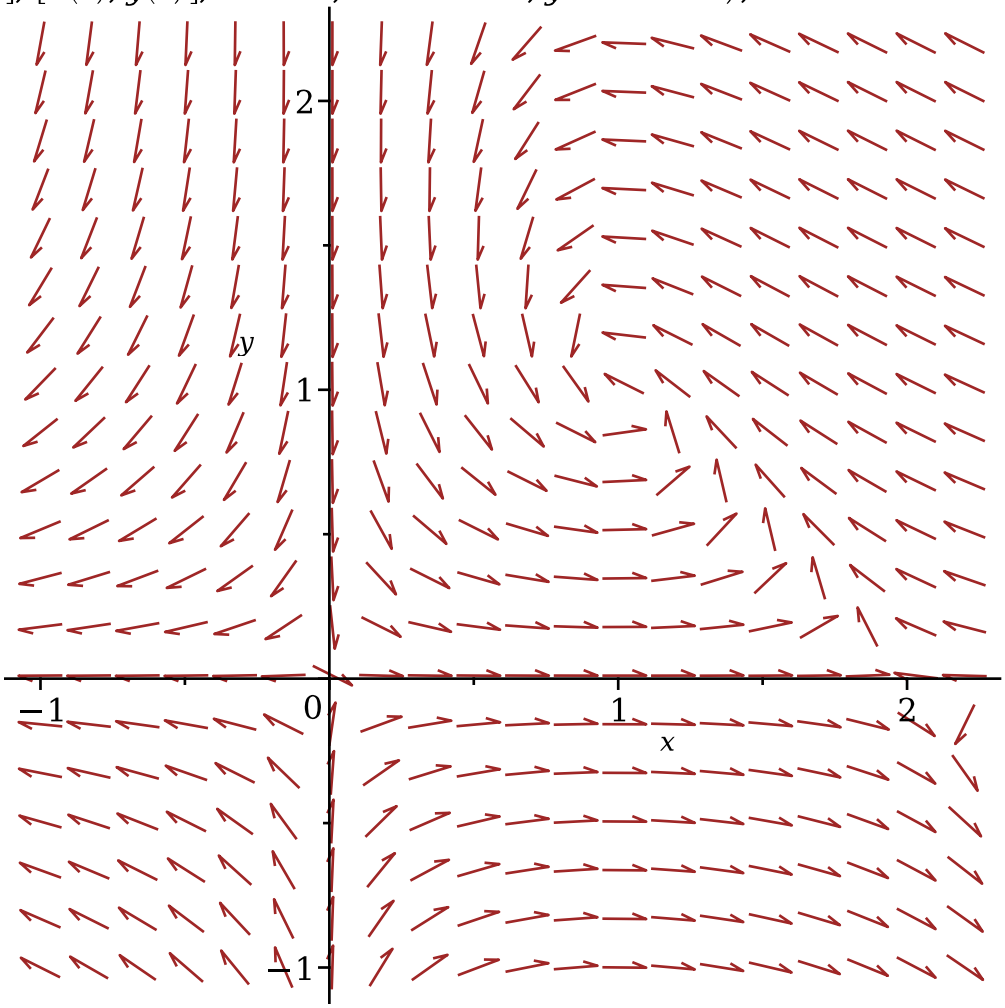
```
> J3 := subs([x = 1, y = 1], Jm);
```

$$J3 := \begin{bmatrix} -1 & -1 \\ 1 & 0 \end{bmatrix} \tag{10}$$

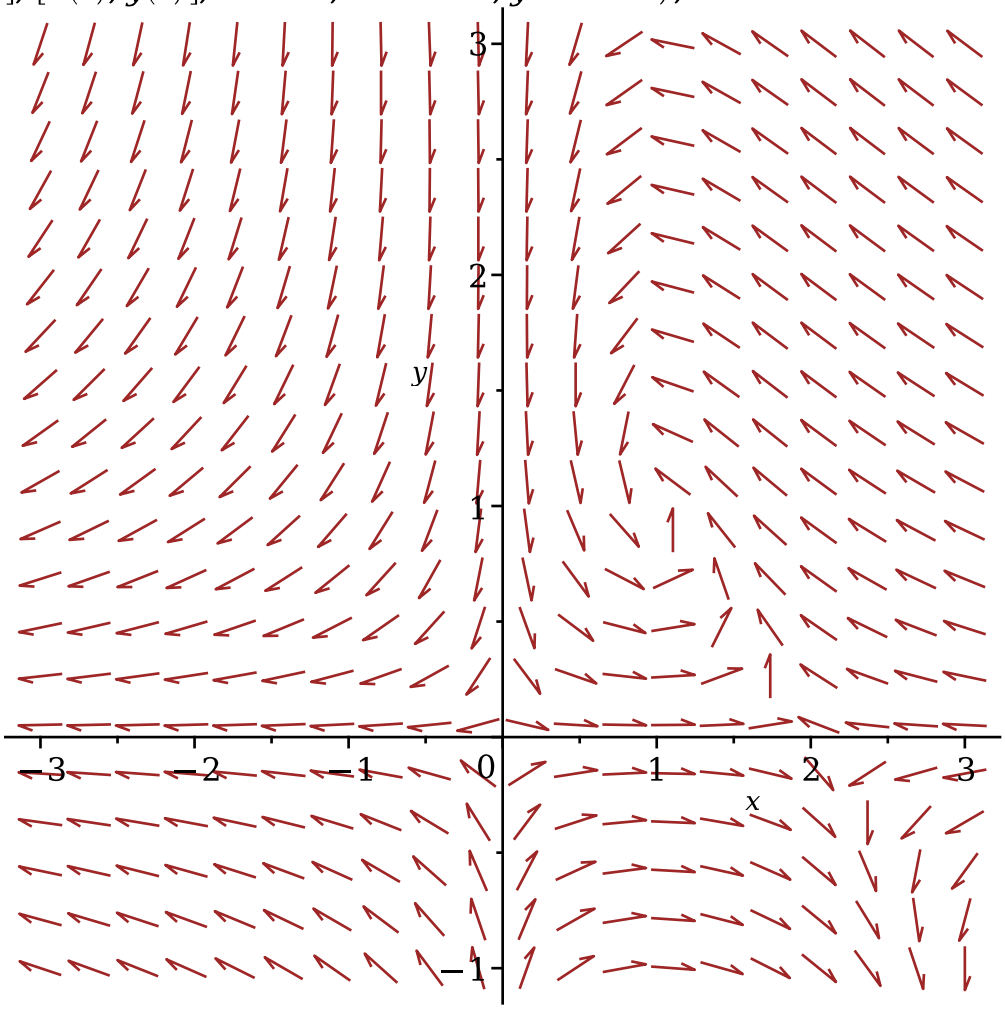
```
> eigenvalues(J3);
```

$$-\frac{1}{2} + \frac{I\sqrt{3}}{2}, -\frac{1}{2} - \frac{I\sqrt{3}}{2} \tag{11}$$

```
> #c)
> dfieldplot([diff(x(t), t) = 2·x(t) − (x(t))^2 − x(t)·y(t), diff(y(t), t) = −y(t) + x(t)
      ·y(t)], [x(t), y(t)], t = 0..1, x = −1..2.2, y = −1..2.2);
```



```
> #d)n am scris tot
> DEplot( [diff(x(t), t) = 2·x(t) − (x(t))^2 − x(t)·y(t), diff(y(t), t) = −y(t) + x(t)
      ·y(t)], [x(t), y(t)], t = 0..1, x = −3..3, y = −1..3);
```




```
>
> #e)
>
>
>
>
>
> #2
> restart;
> eq1 := x − 2·x·y
> eq2 := x^2/2 − y
> solve( {x − 2·x·y = 0, x^2/2 − y = 0}, {x, y});
```

$$\{x = 0, y = 0\}, \left\{x = 1, y = \frac{1}{2}\right\}, \left\{x = -1, y = \frac{1}{2}\right\} \qquad \textbf{(12)}$$

```
>
> #b)
> 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, 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*]

(13)

```
> with(VectorCalculus)
```

[*&x, `*`, `+`, `-`, `.`, <,>, <|>, About, AddCoordinates, ArcLength, BasisFormat, Binormal, ConvertVector, CrossProduct, Curl, Curvature,* D, *Del, DirectionalDiff, Divergence, DotProduct, Flux, GetCoordinateParameters, GetCoordinates, GetNames, GetPVDescription, GetRootPoint, GetSpace, Gradient, Hessian, IsPositionVector, IsRootedVector, IsVectorField, Jacobian, Laplacian, LineInt, MapToBasis,∇, Norm, Normalize, PathInt, PlotPositionVector, PlotVector, PositionVector, PrincipalNormal, RadiusOfCurvature, RootedVector, ScalarPotential, SetCoordinateParameters,*

(14)

SetCoordinates, SpaceCurve, SurfaceInt, TNBFrame, TangentLine, TangentPlane, TangentVector, Torsion, Vector, VectorField, VectorPotential, VectorSpace, Wronskian, diff, eval, evalVF, int, limit, series]

```
> with(linalg)
```

[*BlockDiagonal, GramSchmidt, JordanBlock, LUdecomp, QRdecomp, Wronskian, addcol, addrow, adj, adjoint, angle, augment, backsub, band, basis, bezout, blockmatrix, charmat, charpoly, cholesky, col, coldim, colspace, colspan, companion, concat, cond, copyinto, crossprod, curl, definite, delcols, delrows, det, diag, diverge, dotprod, eigenvals, eigenvalues, eigenvectors, eigenvects, entermatrix, equal, exponential, extend, ffgausselim, fibonacci, forwardsub, frobenius, gausselim, gaussjord, geneqns, genmatrix, grad, hadamard, hermite, hessian, hilbert, htranspose, ihermite, indexfunc, innerprod, intbasis, inverse, ismith, issimilar, iszero, jacobian, jordan, kernel, laplacian, leastsqrs, linsolve, matadd, matrix, minor, minpoly, mulcol, mulrow, multiply, norm, normalize, nullspace, orthog, permanent, pivot, potential, randmatrix, randvector, rank, ratform, row, rowdim, rowspace, rowspan, rref, scalarmul, singularvals, smith, stackmatrix, submatrix, subvector, sumbasis, swapcol, swaprow, sylvester, toeplitz, trace, transpose, vandermonde, vecpotent, vectdim, vector, wronskian*]

```
> Jm := Jacobian([x - 2*x*y, x^2/2 - y], [x, y]);
```

$$Jm := \begin{bmatrix} -2y+1 & -2x \\ x & -1 \end{bmatrix} \quad (16)$$

```
>
> J1 := subs([x = 0, y = 0], Jm);
```

$$J1 := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (17)$$

```
> eigenvalues(J1);
```

$$1, -1 \quad (18)$$

```
> J2 := subs([x = 2, y = 0], Jm)
```

$$J2 := \begin{bmatrix} 1 & -4 \\ 2 & -1 \end{bmatrix} \quad (19)$$

```
> eigenvalues(J2);
```

$$\mathrm{I}\sqrt{7},\ -\mathrm{I}\sqrt{7} \tag{20}$$

> $J3 := subs([x = 1, y = 1], Jm);$

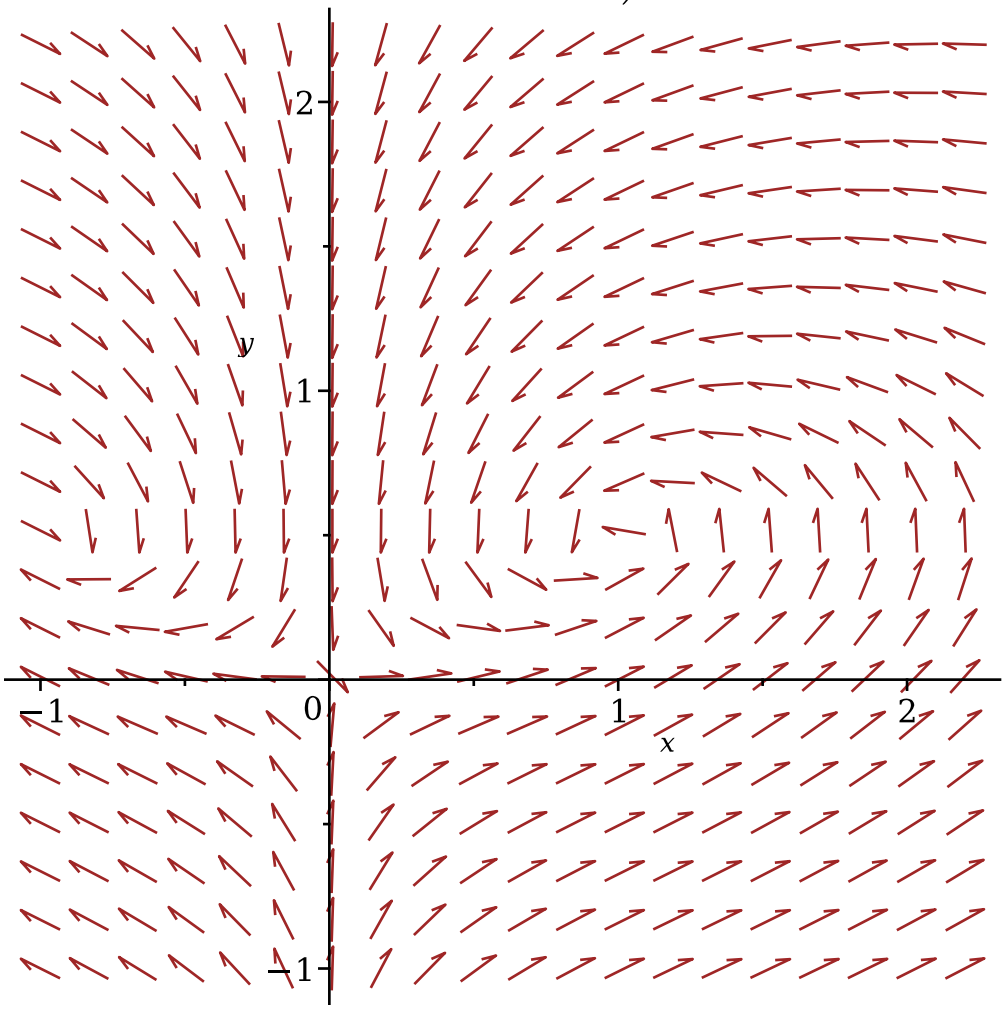
$$J3 := \begin{bmatrix} -1 & -2 \\ 1 & -1 \end{bmatrix} \tag{21}$$

> $eigenvalues(J3);$

$$-1 + \mathrm{I}\sqrt{2},\ -1 - \mathrm{I}\sqrt{2} \tag{22}$$

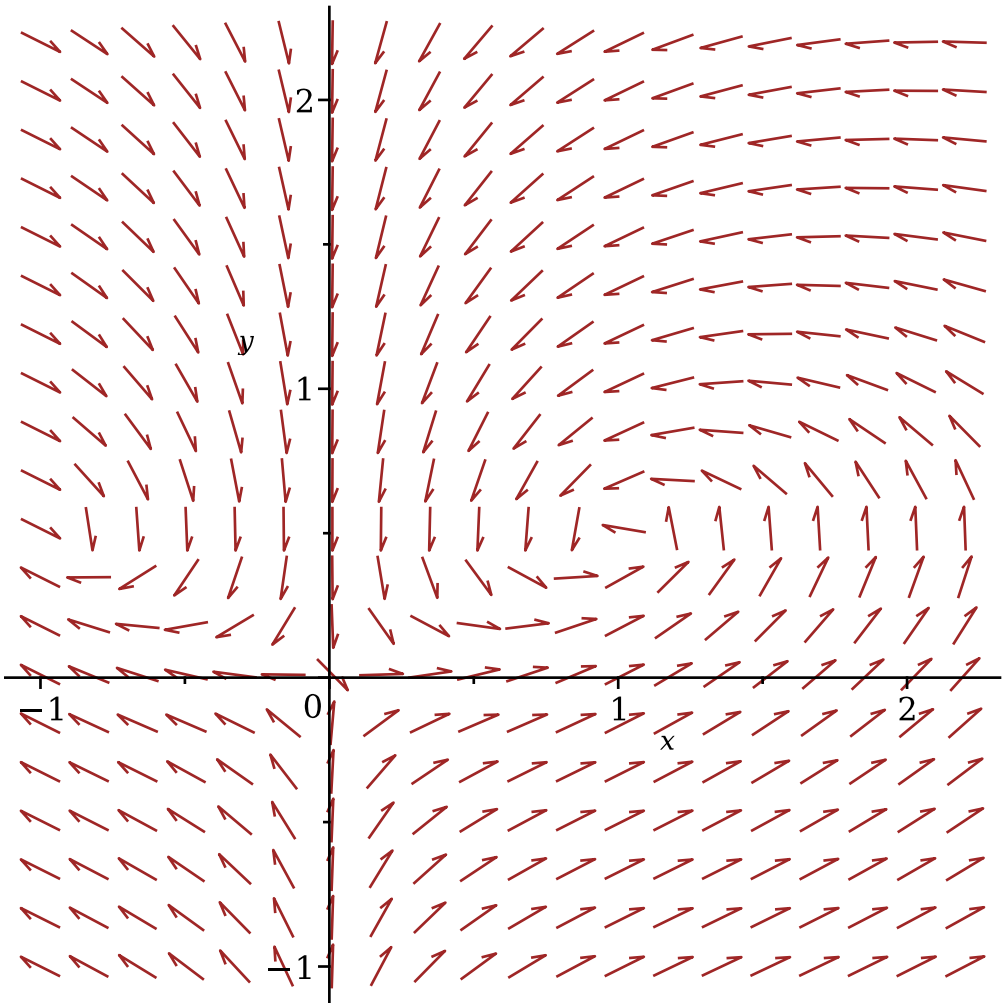
> *#c)*

> $dfieldplot\left(\left[diff(x(t), t) = x(t) - 2\cdot x(t)\cdot y(t),\ diff(y(t), t) = \frac{(x(t))^2}{2} - y(t)\right],\ [x(t), y(t)], t = 0..1, x = -1..2.2, y = -1..2.2\right);$



> *#d)n am scris tot*

> $DEplot\left(\left[diff(x(t), t) = x(t) - 2\cdot x(t)\cdot y(t),\ diff(y(t), t) = \frac{(x(t))^2}{2} - y(t)\right], [x(t), y(t)], t = 0..1, x = -1..2.2, y = -1..2.2\right);$




```
>
>
>
> #----------------------------------------------EX3:
> #a)----------
> restart;
> with(linalg) : with(DEtools) : with(VectorCalculus) :
>
> Jm := Jacobian([y,−4·sin(x)], [x, y]);
```

$$Jm := \begin{bmatrix} 0 & 1 \\ -4\cos(x) & 0 \end{bmatrix} \quad (23)$$

```
> J := subs([x = 0, y = 0], Jm)
```

$$J := \begin{bmatrix} 0 & 1 \\ -4\cos(0) & 0 \end{bmatrix} \quad (24)$$

```
> eigenvalues(J);
```

$$2\,I,\ -2\,I \quad (25)$$

```
> #b)----------
```

```
> dsolve(diff(y(x), x) = -4·sin(x)/y(x), y(x));
```

$$y(x) = \sqrt{8\cos(x) + c_1}\,,\ y(x) = -\sqrt{8\cos(x) + c_1} \qquad (26)$$

```
> H := (x, y) → y^2 - 8·cos(x);
```

$$H := (x, y) \mapsto y^2 + (-8\cdot\cos(x)) \qquad (27)$$

```
> H(x, y);
```

$$y^2 - 8\cos(x) \qquad (28)$$

```
> #c)----------
```

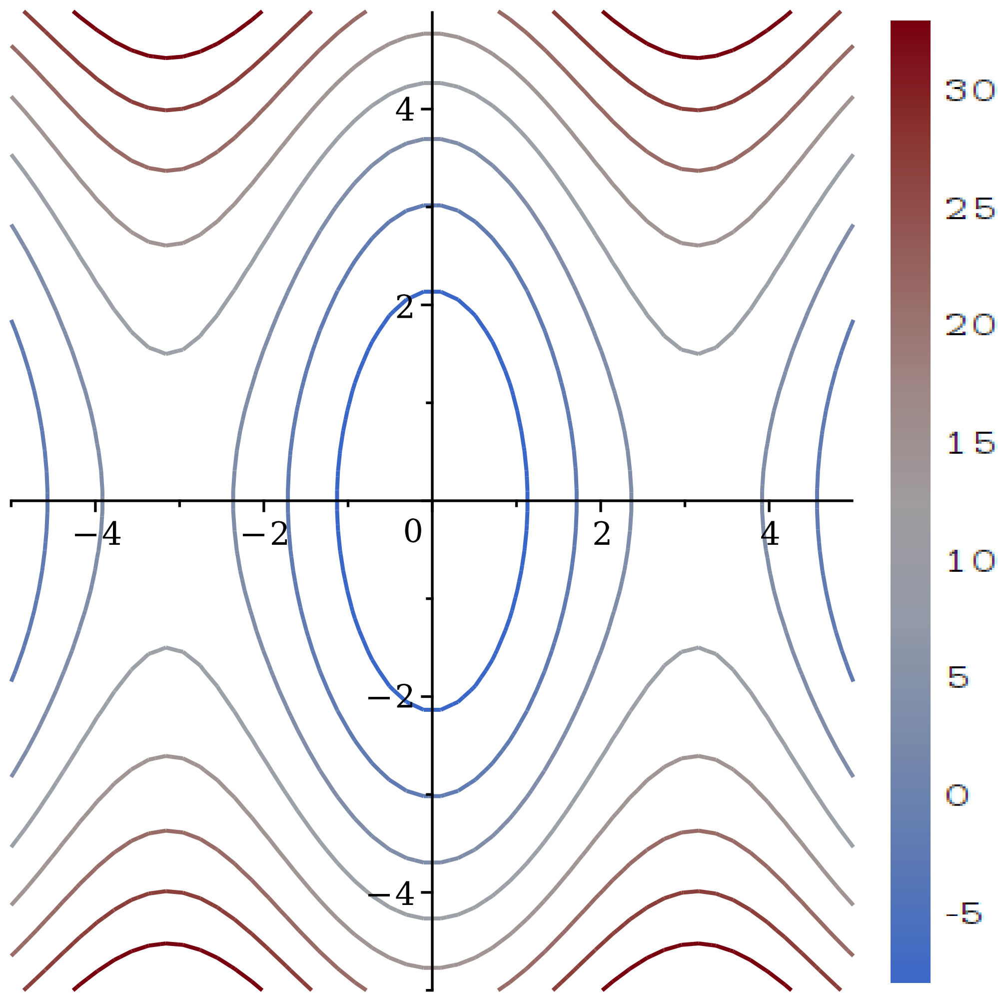
```
> with(plots)
```

[*animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, densityplot, display, dualaxisplot, fieldplot, fieldplot3d, gradplot, gradplot3d, implicitplot, implicitplot3d, inequal, interactive, interactiveparams, intersectplot, listcontplot, listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple, odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d, polyhedra_supported, polyhedraplot, rootlocus, semilogplot, setcolors, setoptions, setoptions3d, shadebetween, spacecurve, sparsematrixplot, surfdata, textplot, textplot3d, tubeplot*] (29)

```
> diff(H(x, y), x)·y-diff(H(x, y), y)·4· sin(x)
```

$$0 \qquad (30)$$

```
> contourplot(H(x, y), x = -5..5, y = -5..5)
```



>

>

>

> #--EX4

> *similar cu 3 tema 4 , aproape la fel*

>

>

>

>

>

>

>

>

>

>

>

>

>

>