Міністерство освіти і науки України
ЧЕРКАСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ім. Богдана Хмельницького

Факультет Обчислювальної техніки, інтелектуальних та управляючих систем
Кафедра Програмного забезпечення автоматизованих систем

Практична робота № 2

по дисципліні **«Диференційні рівняння»**

Тема: Побудова інтегральних кривих за допомогою ізоклін

Варіант 14

Виконала: студент гр. КС-231
Попов А.А.

Перевірив: старший викладач
кафедри ПЗАС
Гук В.І.

Черкаси, 2025

Теоретичні відомості:

Пакет розв'язання диференціальних рівнянь **DEtools**.

Побудова поля напрямків оператором **dfieldplot**

Оператор **DEplot** *для візуалізації інтегральних кривих і побудови поля напрямків.*

Побудова ізоклін як графіків неявних функцій операторами **implicitplot** *та* ***contourplot*** *пакета* **plots.**

2.6.3. Інструментальний пакет розв'язування диференціальних рівнянь DEtools

Пакет DEtools містить специфічні засоби для аналітичного та числового розв'язування диференціальних рівнянь та їх систем, візуалізації розв'язку різними способами.

Звертатися до команд пакета можна двома стандартними способами:

```
> DEtools[command](arguments);
```

```
> command(arguments).
```

Розглянемо найбільш важливі функції із цього пакета.

- **autonomous** (тестує диференціальне рівняння на предмет автономності, тобто випадку, коли незалежна змінна до рівняння не входить в явному вигляді);
- **convertsys** (перетворює систему диференціальних рівнянь в систему першого порядку);
- **reduceOrder** (забезпечує зниження порядку диференціального рівняння);
- **regularsp** (знаходить особливі точки неавтономного лінійного диференціального рівняння першого порядку);
- **varparam** - (розв'язує диференціальне рівняння або систему методом варіації параметрів);
- **de2diffop** (перетворює диференціальне рівняння на диференціальний оператор);

- **diffop2de** (перетворює диференціальний оператор на диференціальне рівняння);

- **DEplot** (будує 2D-розв'язок рівняння або системи);

- **DEplot3d** (будує 3D-розв'язок системи рівнянь);

- **dfieldplot** (будує поле напрямків);

- **PDEplot** (будує розв'язок диференціального рівняння в частинних похідних першого порядку);

- **phaseportrait** (будує фазовий портрет системи диференціальних рівнянь).

Функції з цього пакета для візуалізації розв'язків розглянемо в наступному пункті.

2.6.4. Графічна візуалізація розв'язків диференціаль- них рівнянь

2.6.4.1. Функція **plots [odeplot]**

Для звичайного графічного 2D- або 3D-подання результатів розв'язання диференціальних рівнянь використовується функція odeplot 3 пакета plots. Її синтаксис є такий: odeplot(s, vars, r, o), де s - результат роботи команди dsolve(numeric); vars - змінні; r - - параметр, що задає границі розв'язку, наприклад a..b; o - необов'язкові додаткові опції.

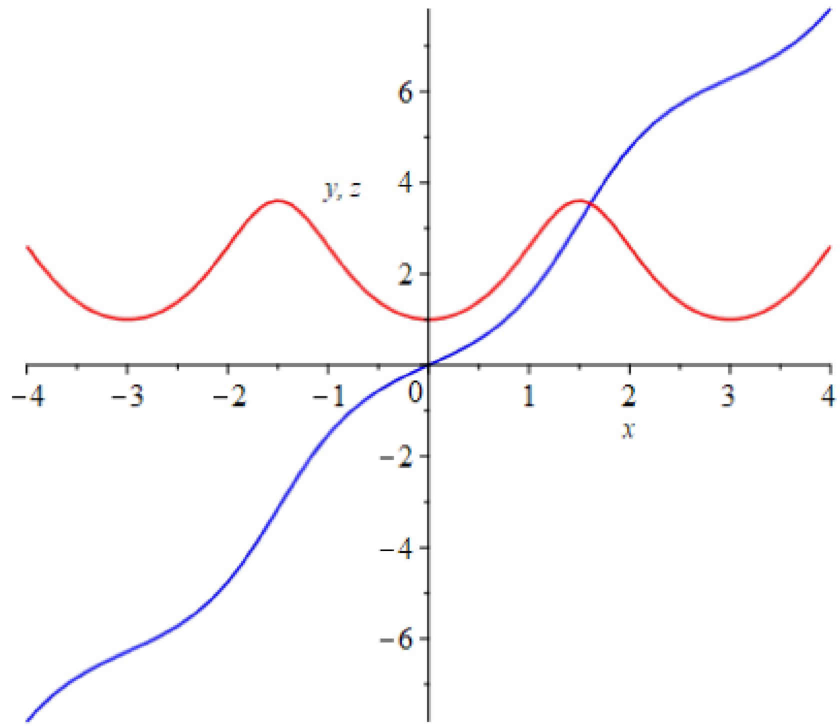
Звичайний розв'язок, як правило, більш наглядний, ніж фазові портрети. Але для спеціалістів (наприклад, у теорії коливань) фазовий портрет дає більше інформації. Він більш трудомісткий для побудови, але система Maple може будувати і фазові портрети.

Приклад. ***Побудувати графіки функцій y(x) та z(x), які є розв'язками системи диференціальних рівнянь, в звичайному вигляді та у вигляді фазового портрета:***

```
>with(plots);

 >sys := diff(y(x), x) = z(x), diff(z(x), x) =
3*sin(y(x));

 >functions := {y(x), z(x)};

 >Incond := y(0) = 0, z(0) = 1;

 > p := dsolve({Incond, sys}, functions, numeric);
```

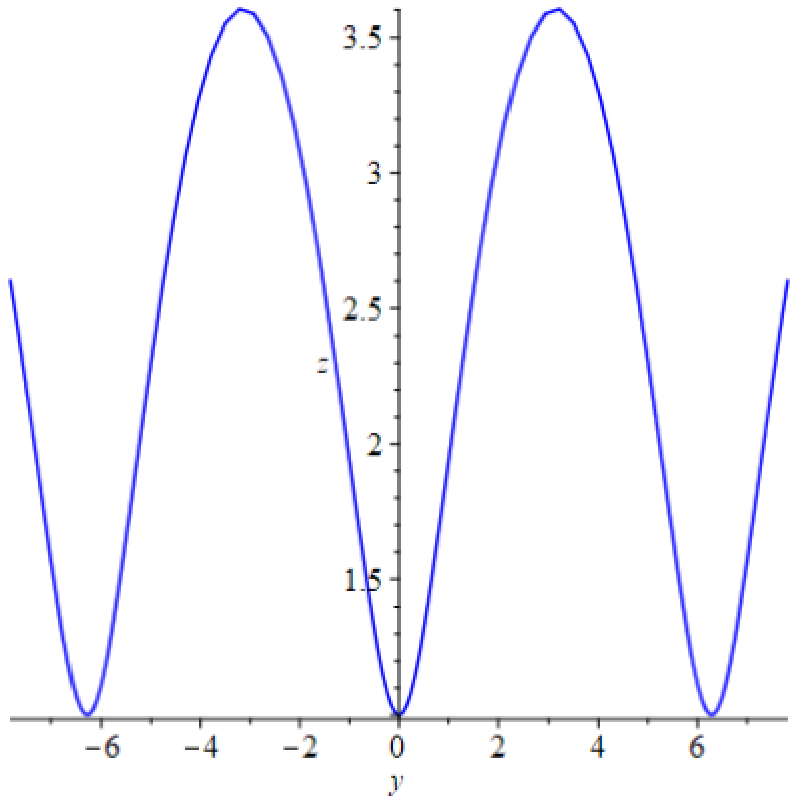
Звичайний графік:

```
>odeplot(p, [[x, y(x)], [x, z(x)]], -4 .. 4, numpoints =
100, color = [blue, red]);
```



Фазовий портрет:

```
>odeplot(p, [y(x), z(x)], -4 .. 4, numpoints = 100, color
= [blue, red]);
```



2.6.4.2. Функція `DEtools[DEplot]`

Розглянемо роботу команди `DEplot` із пакета **DEtools**. Вона чисельно розв'язує диференціальні рівняння та їх системи при одній незалежній змінній та будує графік розв'язку. При цьому використовується метод Рунге-Кутта 4-го порядку, а графічні побудови являють собою або криві, або векторні поля напрямків. Синтаксис функції такий: `DEplot (sde, vars, trange, inits, xrange, yrange, options)`, де `sde` диференціальне рівняння або їх система у фігурних або квадратних дужках; `vars` - залежні змінні у фігурних або квадратних дужках; `trange` — область зміни незалежної змінної `inits` - початкові умови для розв'язання; `xrange` та `yrange` - необов'язкові параметри, що задають область зміни для першої та другої залежної змінної; `options` - опції у вигляді - `keyword=value` («ключове слово = значення»). Опції, які можуть Використовуватися з функцією `DEplot`, наведено в таблиці 2.5.

Таблиця 2.5 - Опції функції **DEplot** із пакета **DEtools** у форматі **keyword=value** («ключове слово = значення»)

Ключове слово	Значення	Зміст оції
animatecurves	true або false	Створює анімацію фазової траєкторії в часі
animatefield	true або false	Створює анімацію поля напрямків
arrows	'small', 'smalltwo', 'medium', 'mediumfill', 'large', 'curve', 'comet', 'line', або 'none'	Задає тип стрілки векторного поля
color	name, RGB або HUE	Задає колір стрілок одним 5 способів: ім'я кольору (name), цифрове позначення за шкалою |RGB або HUE, математичним виразом або процедурою (про колір див. п. 4.1.2)
dirfield	[int,int],int або [[int,int], [ ],...]	Задає координати точок, куди поміщати стрілки (сіткою або кожну стрілку окремо)

iterations	int	Представляє метод для зменшення кроку stepsize при фіксованій кількості точок (int - натуральне число)
linecolor	name	Задає колір лінії
numframes	int	Вводить кількість кадрів при анімації
numpoints	int	Задає кількість точок, якими будується графік
numsteps	int	Задає кількість кроків при обчисленнях (використовується далі опцією stepsize)
obsrange	true або false	Задає, чи прибирати стрілки, що виходять за межі побудови (наприклад, при асимптотичній поведінці в анімації)
scene	[name, name]	Вказує імена залежних змінних, для яких будується графік
size	magnitude або float	Задає розмір стрілок, який визначається або пропорційно величині поля, або заданим числом типу float (за замовчуванням size = 1.0)

stepsize	real	Задає розмір кроку для числового методу обчислення розв'язку рівняння (за замовчуванням stepsize=$^{a-b}/_{numsteps}$ при trange=a..b)

Побудову інтегральної кривої зручно виконувати за допомогою ізоклін. ***<u>Ізокліною</u>*** **називається множина точок $(x; y) \in D$, у яких дотичні до шуканих інтегральних кривих мають однаковий напрямок, тобто виконується умова**

$$\boldsymbol{y' = f(x,y) = k}$$

де $\boldsymbol{k}$ - стала.

Отже, ізокліни диференціального рівняння - це лінії рівня функції $f(x,y)$. Надаючи сталій k різних числових значень, отримаємо сукупність ізоклін, за допомогою яких наближено будуємо інтегральні криві диференціального рівняння.

Зауважимо, що "нульова ізокліна" $f(x,y) = 0$ - рівняння лінії, на якій можуть знаходитися точки екстремумів інтегральних кривих.

Доповнити уяву про вигляд інтегральних кривих можна також, знаючи їх точки перегину. Множину можливих точок перегину інтегральних кривих знаходимо з умови $y'' = 0$. Диференціюючи початкове диференціальне рівняння за змінною x, знаходимо рівняння множини точок області D, які можуть бути точками перегину інтегральних ліній:

$$y'' = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} f(x,y) = 0$$

Домашнє завдання

Задача 1. Складти диференціальне рівняння для заданого сімейства кривих

Варіант 14

14) $x^3 = C\,(x^2 - y^2$

Перепишемо диференціальне рівняння:

$x^3 = C(x^2 - y^2)$

продиференціюєм обидві частини рівняння :

$$\frac{d(x^3)}{dx} = \frac{d(C(x^2 - y^2))}{dx}$$

Ліва частина :

$$\frac{d(x^3)}{dx} = 3\,x^2$$

Права частина :

$$\frac{d(C(x^2 - y^2))}{dx} = C\left(2\,x - 2\,y\frac{dy}{dx}\right)$$

$$3\,x^2 = C\left(2\,x - 2\,y\frac{dy}{dx}\right)$$

виразимо С з *отриманого рівняння* :

$$C = \frac{3\,x^2}{2\,x - 2\,y\frac{dy}{dx}}$$

Підставимо С в *початкове рівняння* :

$$x^3 = \frac{3\,x^2}{2\,x - 2\,y\frac{dy}{dx}}\,(x^2 - y^2)$$

$$x^3 \cdot \left(2\,x - 2\,y\frac{dy}{dx}\right) = 3\,x^2 \cdot (x^2 - y^2)$$

Диференціальне рівняння складено. Виконаємо перевірку в *Maple* :

> $ODE := \dfrac{3 \cdot x^2}{2 \cdot x - 2 \cdot y(x) \cdot diff(y(x), x)} \cdot (x^2 - y(x)^2) = x^3$

$$ODE := \frac{3\,x^2\,(x^2 - y(x)^2)}{2\,x - 2\,y(x)\left(\frac{d}{dx}\,y(x)\right)} = x^3 \qquad \textbf{(1)}$$

> $dsolve(ODE, y(x))$

Warning, it is required that the numerator of the given ODE depends on the highest derivative. Returning NULL.

$$y(x) = \sqrt{c_1 x + 1}\, x, y(x) = -\sqrt{c_1 x + 1}\, x \quad \textbf{(2)}$$

> *restart*

>

Задача 2. Для заданих диференціальних рівнянь побудувати не менше 5 ізоклін. Побувати векторне поле напрямків для заданого диференціального рівняння.

Варіант 14

14) $xy' - y = 0$

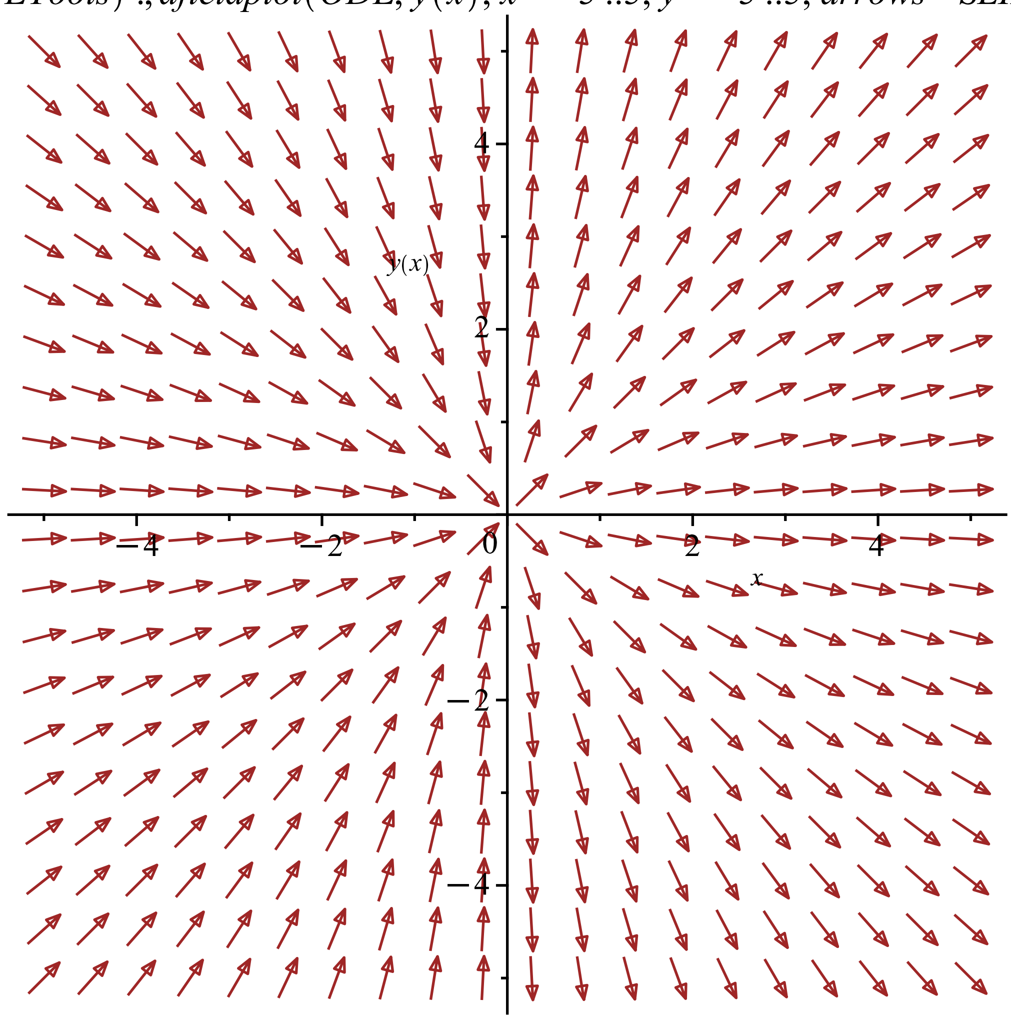
> $ODE := x \cdot diff(y(x), x) - y(x) = 0$

$$ODE := x \left(\frac{d}{dx} y(x) \right) - y(x) = 0 \quad (3)$$

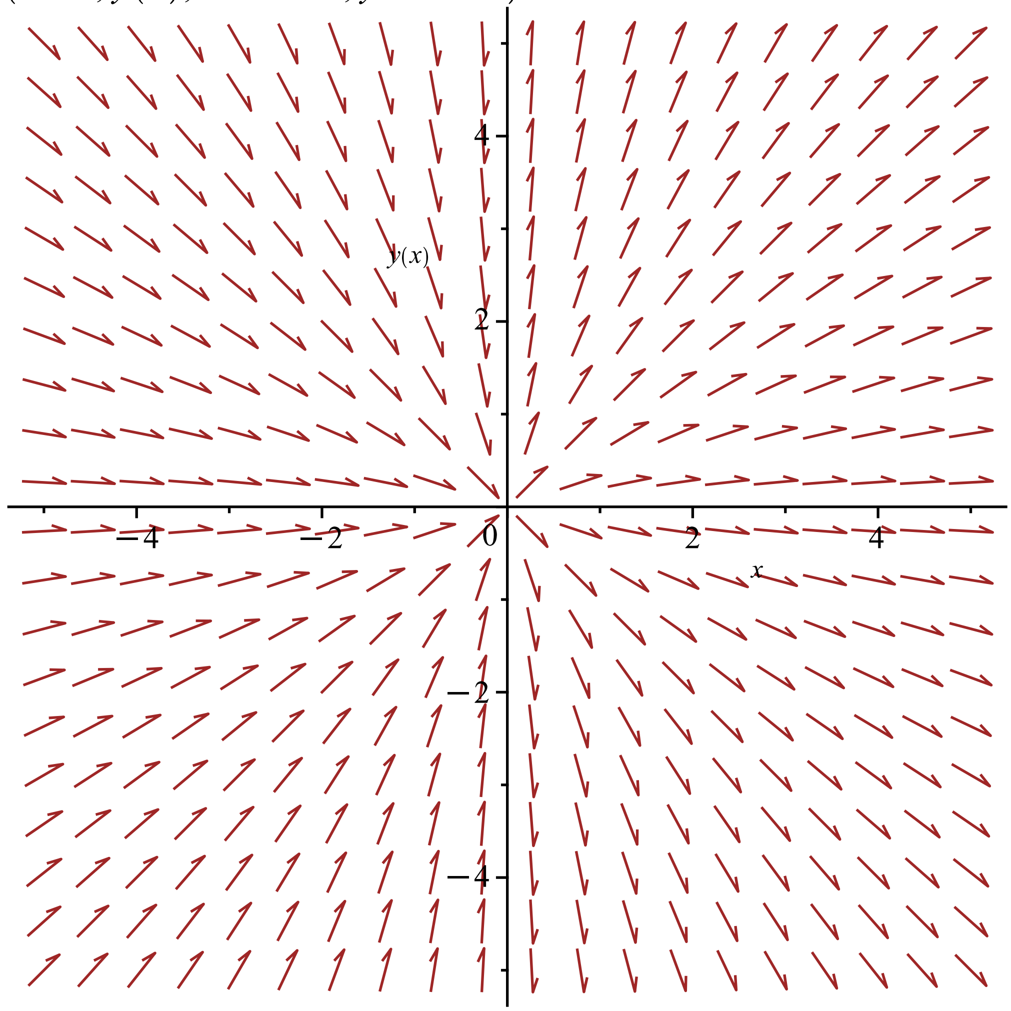
>

Побудуємо поле напрямків скориставшись методвми dfieldplot та DEPlot з *пакету DETools* :

> $with(DETools) ::; dfieldplot(ODE, y(x), x = -5 ..5, y = -5 ..5, arrows = SLIM)$



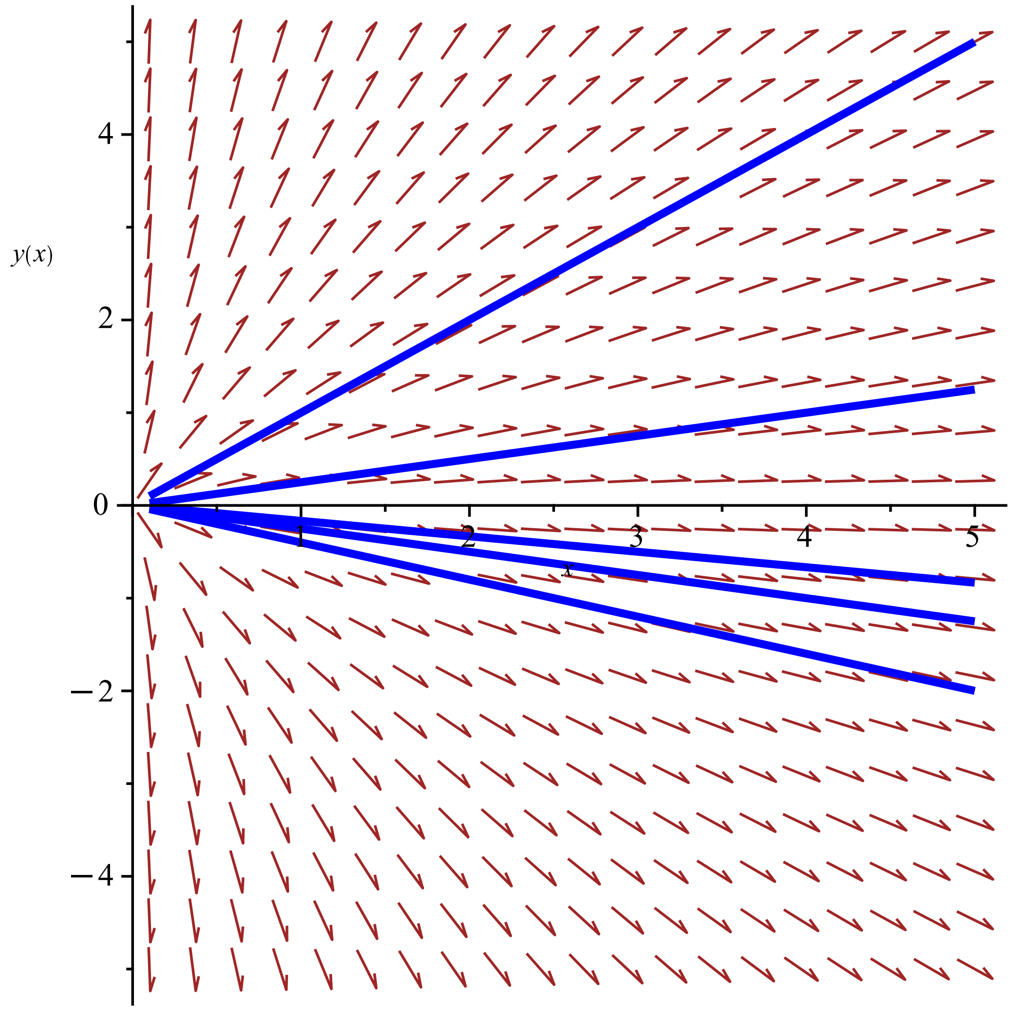
> $DEplot(ODE, y(x), x=-5..5, y=-5..5)$



>

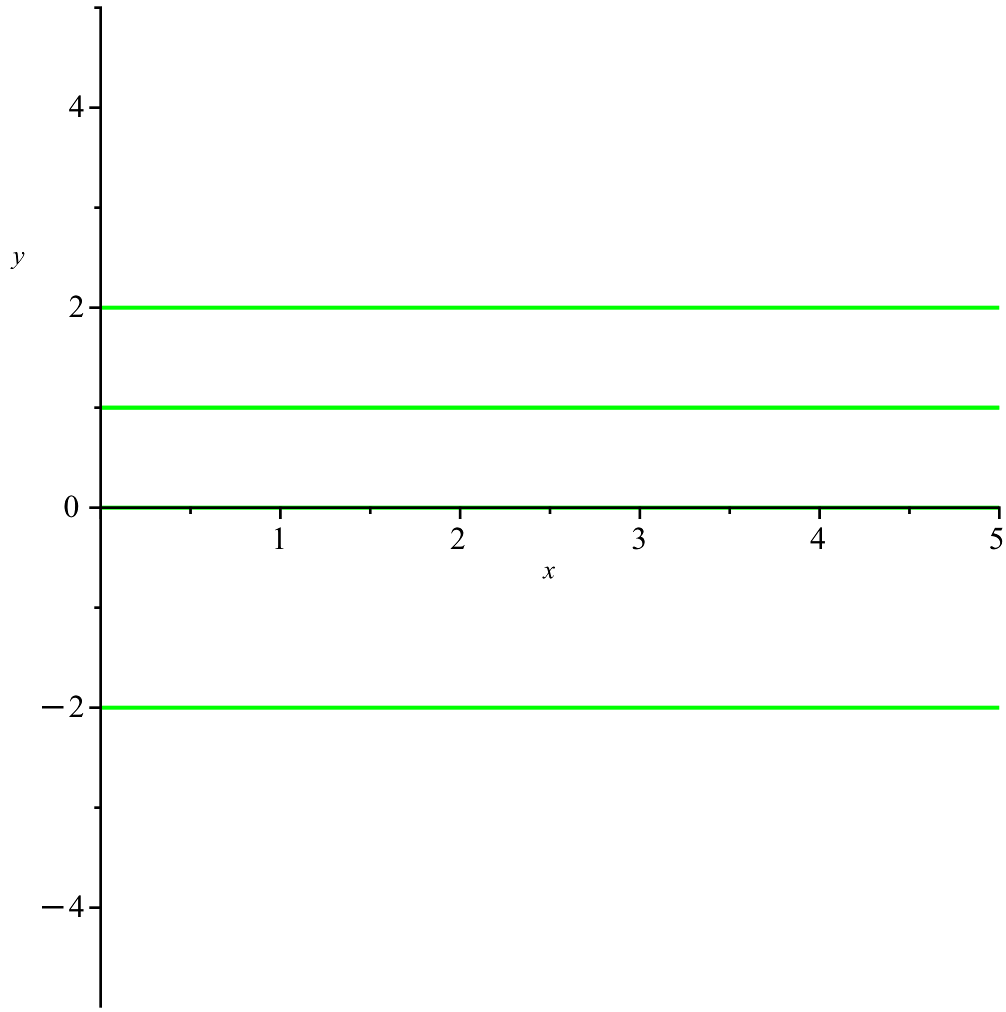
DEplot дозволяє нанести на графік лінії рівня,
які визначаються початковою умовою.

> $plot1 := DEplot(ODE, y(x), x=0.1..5, y=-5..5, [y(1)=1, y(2)=0.5, y(3)=-0.5, y(4)=-1, y(5)=-2], linecolor=blue);$



> *plot2* := *plot*([−2, 1, 0, 1, 2], *x* = 0 ..5, *y* = −5 ..5, *color* = *green*)

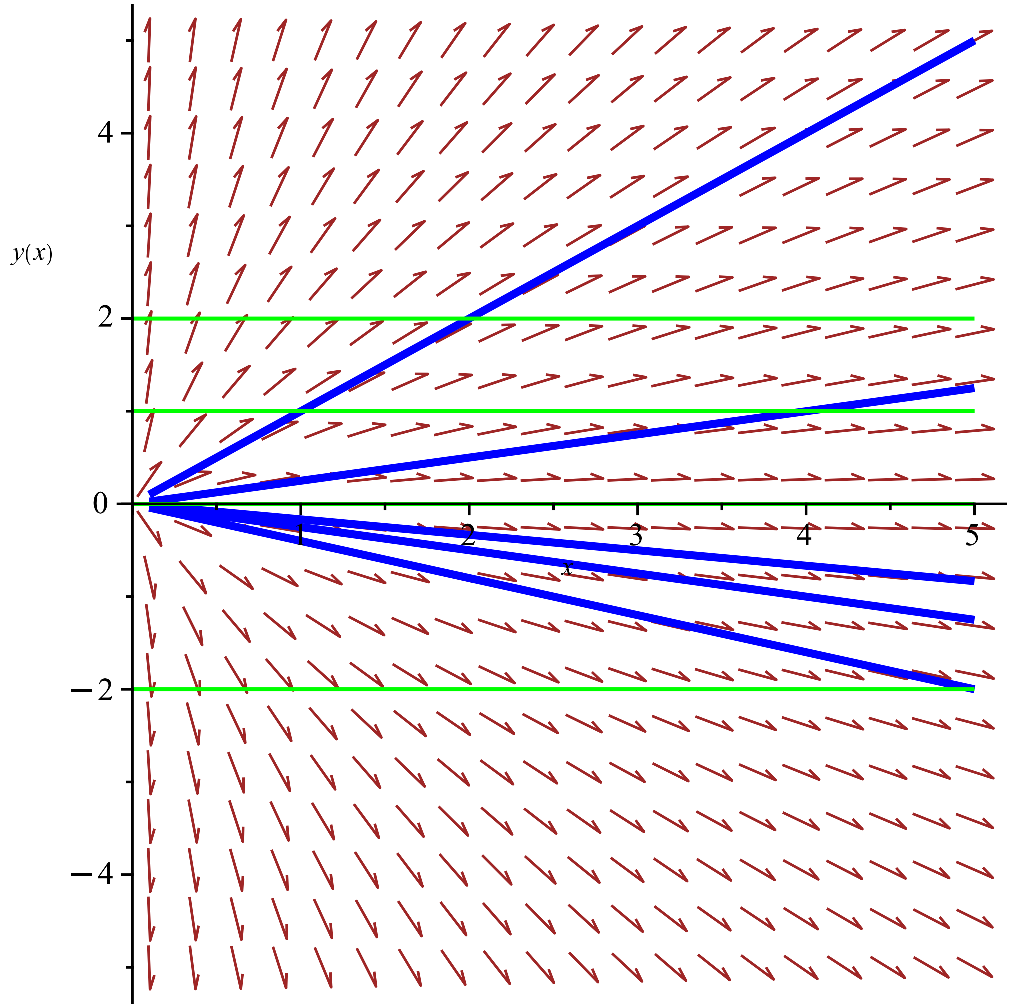
На готовий графік потрібно нанести не менше п'*яти ізоклін.*



> $with(plots)$

[*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*] **(4)**

> $display(plot1, plot2)$



>
Завдання виконано.

> *restart*

>

Задача 3. Для заданих диференціальних рівнянь побудувати ізокліни, побудувати векторне поле напрямків та побудувати інтегральні криві, які проходять через вказані точки

Варіант 14%12 = 2

2) а) $y' = y + (x+1)^2,$	(0; 0),	(0; -2);
б) $y' = x - 1 + Iny,$	(0; 1),	(0; 3);
в) $y' = arctgy + x,$	(0; 0),	(0; -2);
г) $y' = x^2 + (y-1)^2,$	(0; 0),	(0; 3).

Запишемо рівняння а):

$a)\ y` = y + (x+1)^2$

> $ODE := diff(y(x), x) = y + (x+1)^2$

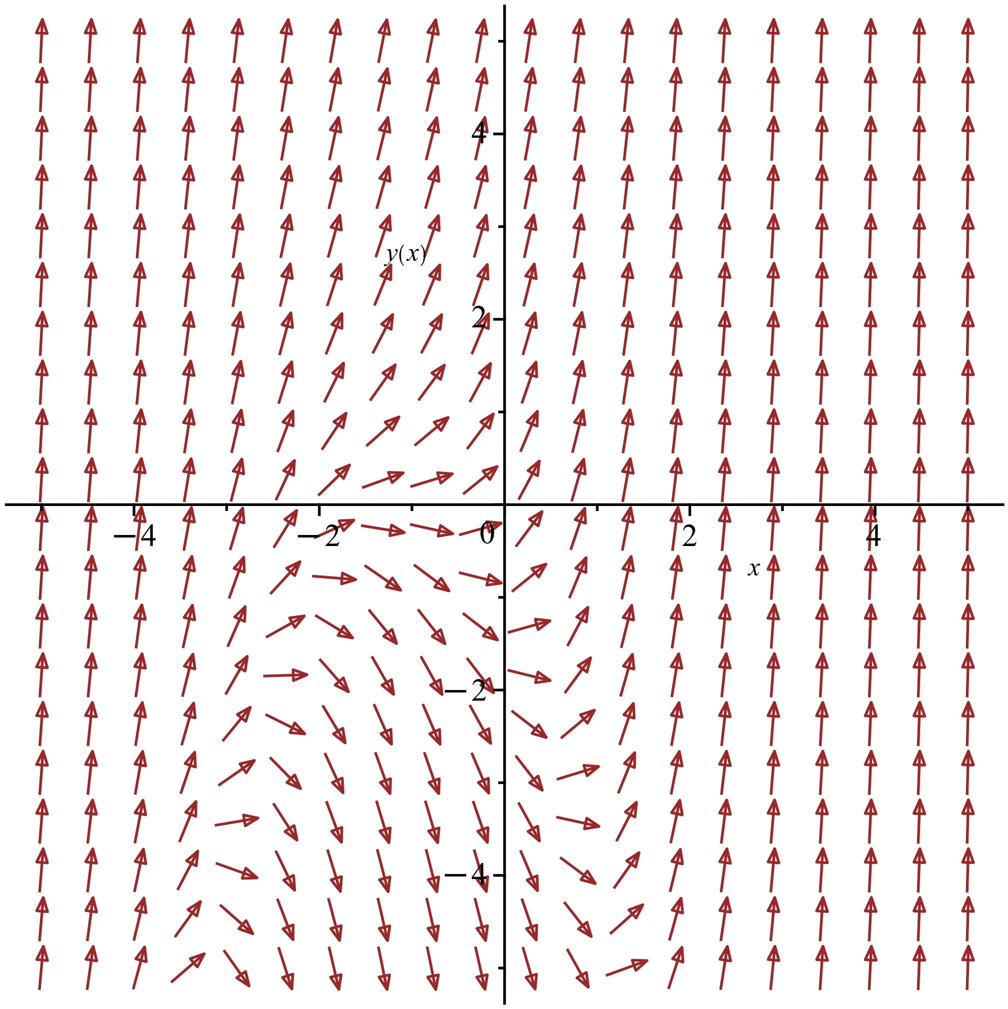
$$ODE := \frac{d}{dx}\, y(x) = y + (x+1)^2 \quad (5)$$

> $with(DEtools)$

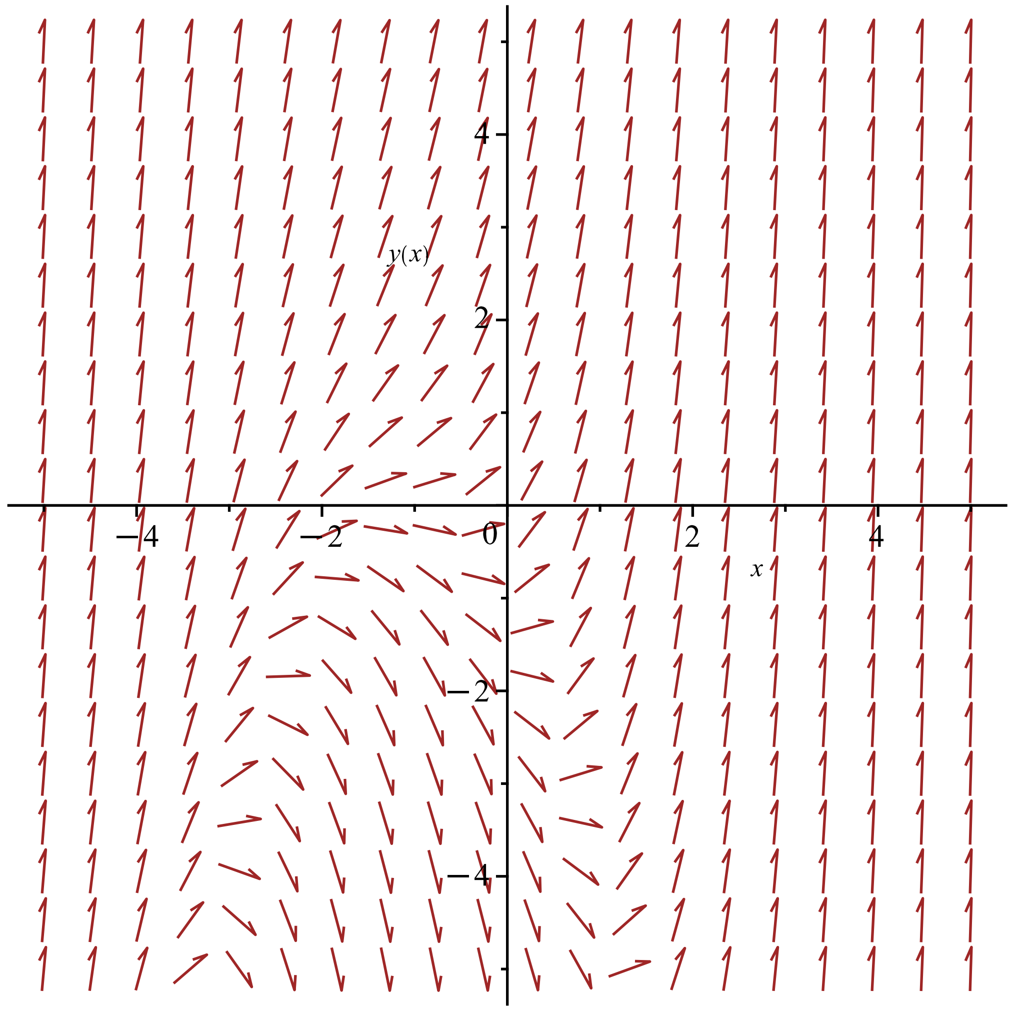
Будуємо поле напрямків :

[*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*, **(6)** *symmetric_power, symmetric_product, symtest, transinv, translate, untranslate, varparam, zoom*]

> $dfieldplot(ODE, y(x), x = -5 ..5, y = -5 ..5, arrows = SLIM)$

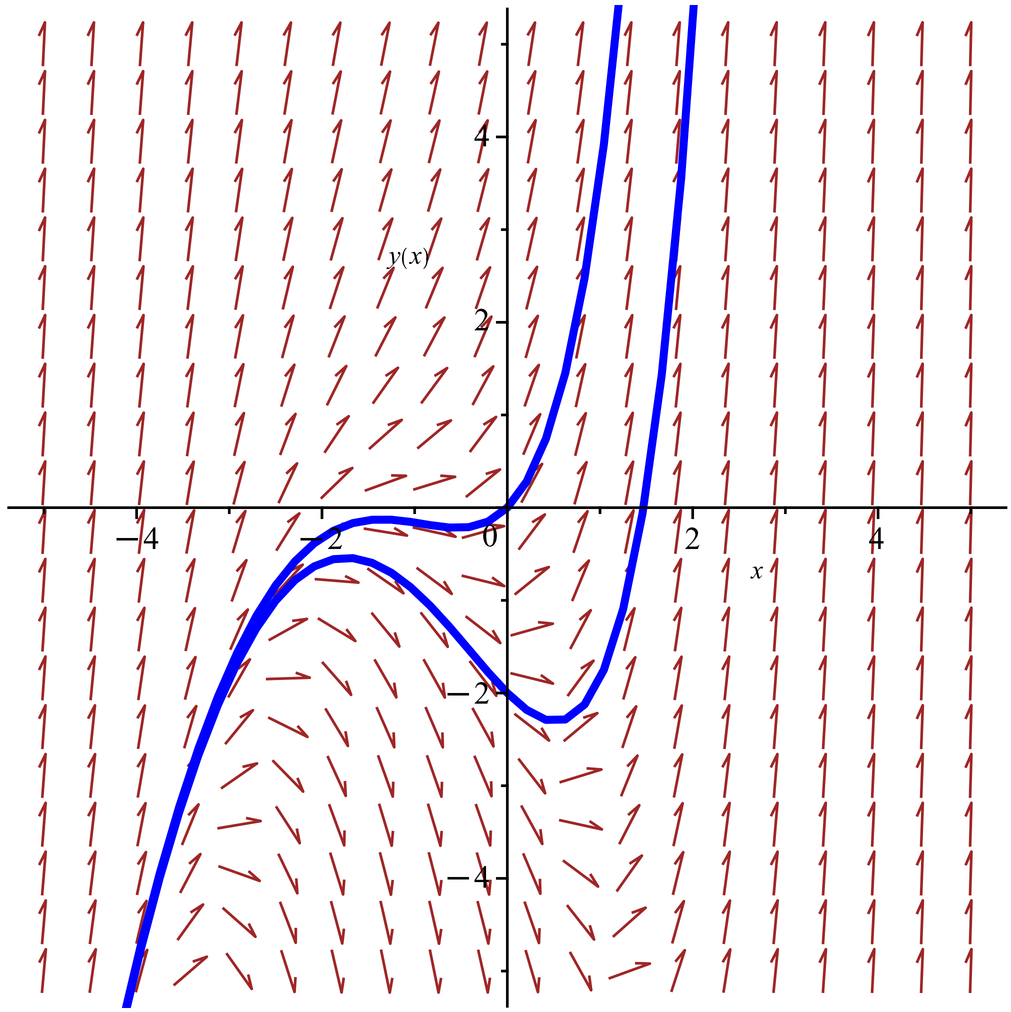


> $DEplot(ODE, y(x), x = -5 ..5, y = -5 ..5)$



> $plot1 := DEplot(ODE, y(x), x = -5..5, y = -5..5, [y(0) = 0, y(0) = -2], linecolor = blue)$

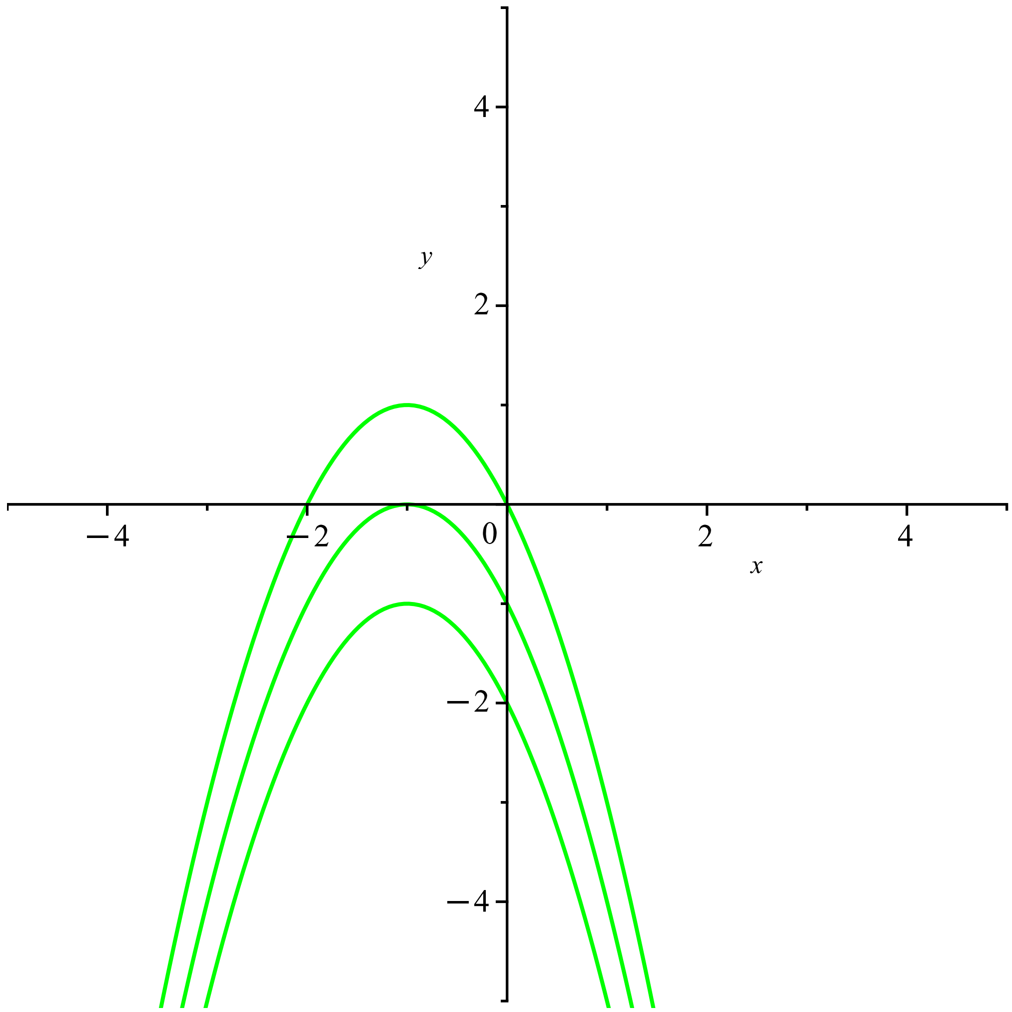
Будуємо інтегральні криві за початковою умовою :



>

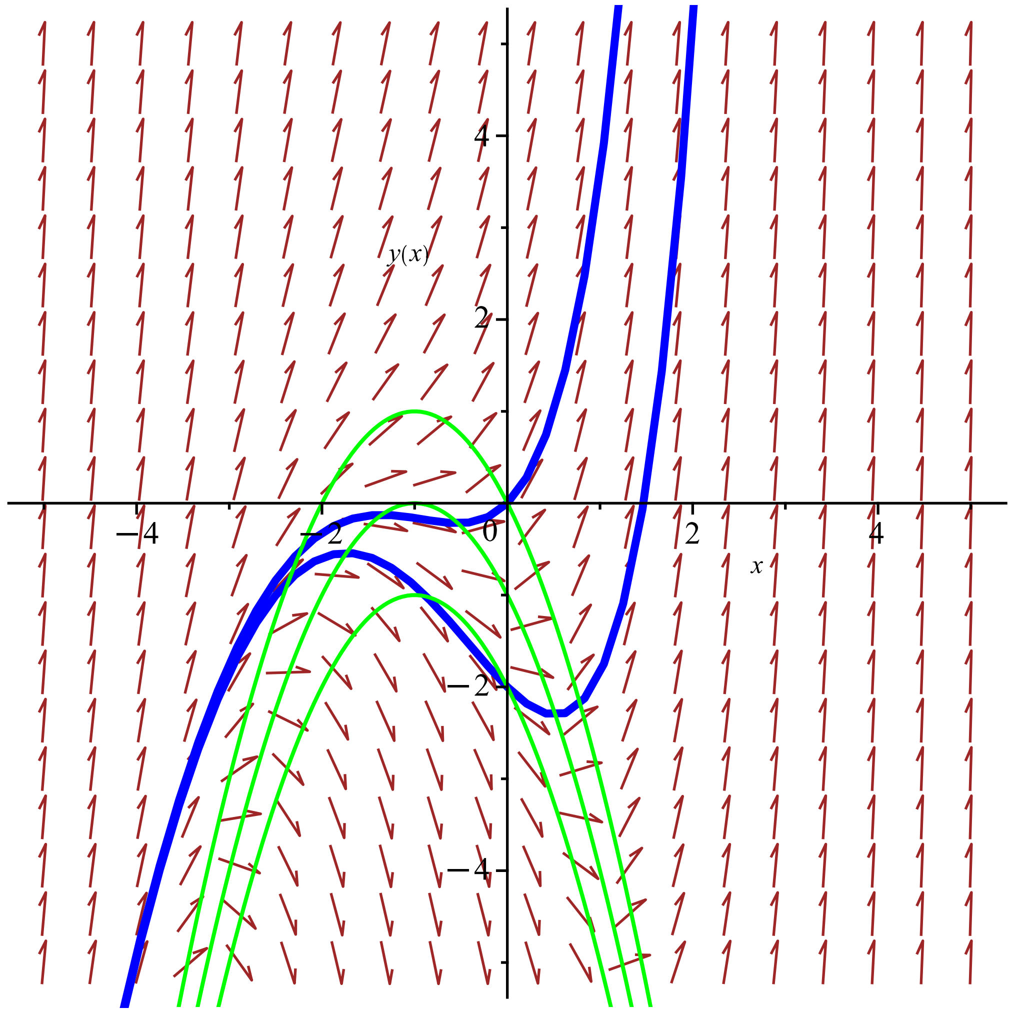
$y = C - (x+1)^2$

> $plot2 := plot([-1-(x+1)^2, -(x+1)^2, 1-(x+1)^2], x = -5..5, y = -5..5, color = green)$



> $with(plots)$:;

> $display(plot1, plot2)$



>

б) $y' = x - 1 + \ln y,$ (0; 1), (0; 3);

Ізоклін:

$\ln(y) = C - (x - 1)$

$y = \mathrm{e}^{C-(x-1)}$

$y = \mathrm{e}^{C} \cdot \mathrm{e}^{(1-x)}$

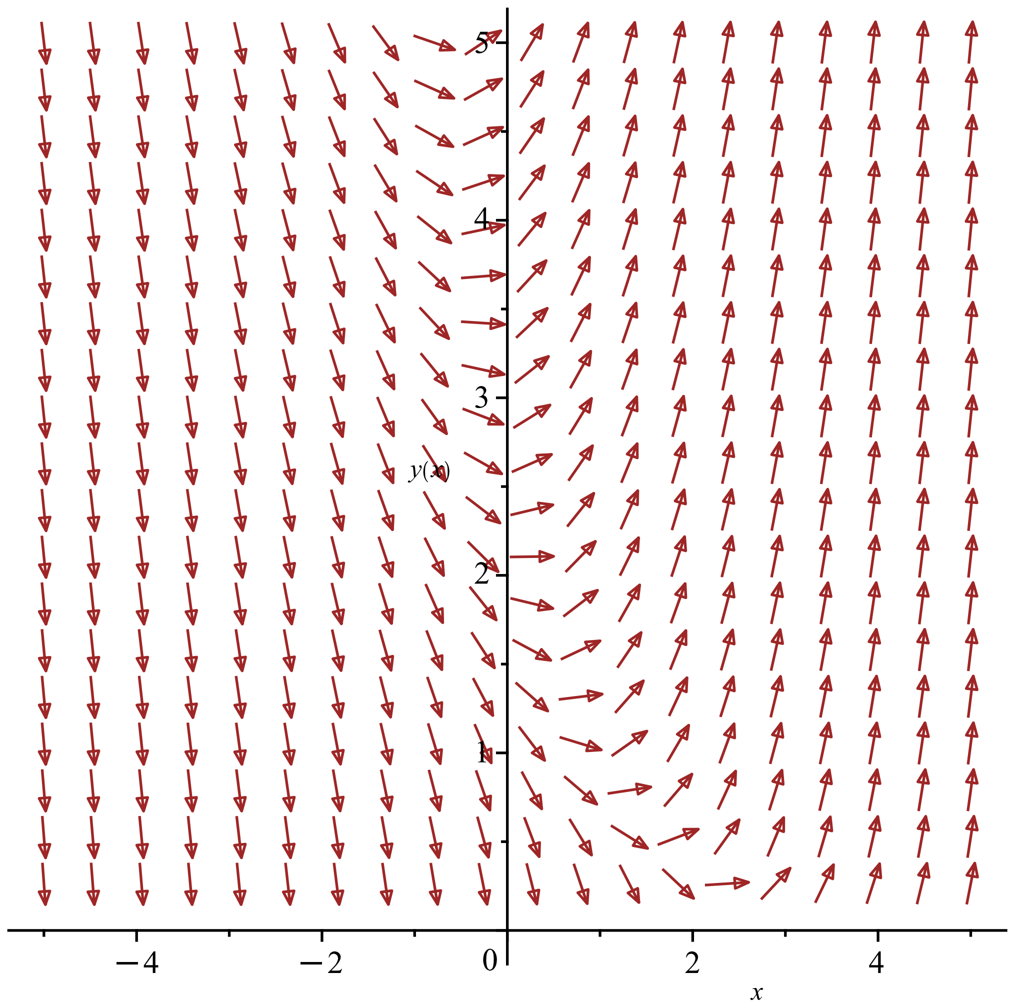
$y = C \cdot \mathrm{e}^{(1-x)}$

> $ODE := diff(y(x), x) = x - 1 + \ln(y(x))$

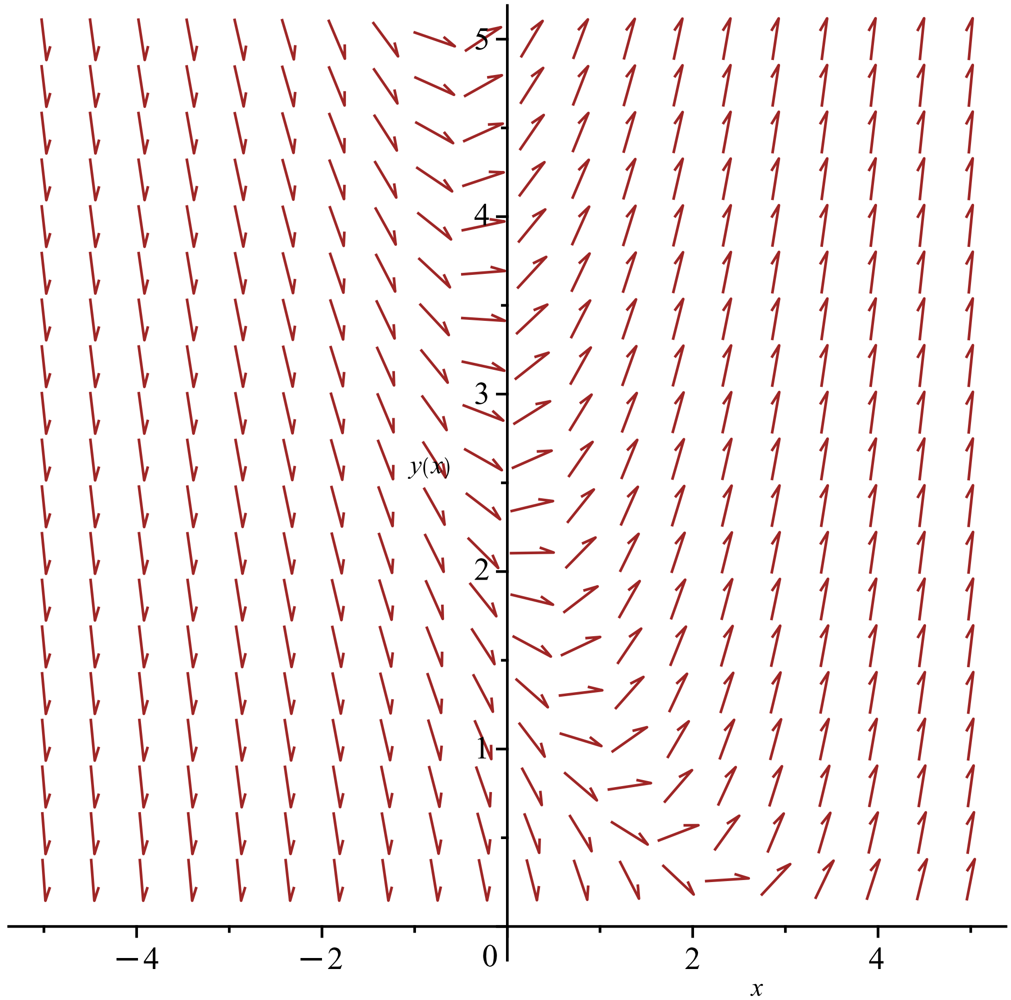
$$ODE := \frac{\mathrm{d}}{\mathrm{d}x} y(x) = x - 1 + \ln(y(x)) \tag{7}$$

> $dfieldplot(ODE, y(x), x = -5..5, y = 0..5, arrows = SLIM)$

Будуємо поле напрямків :



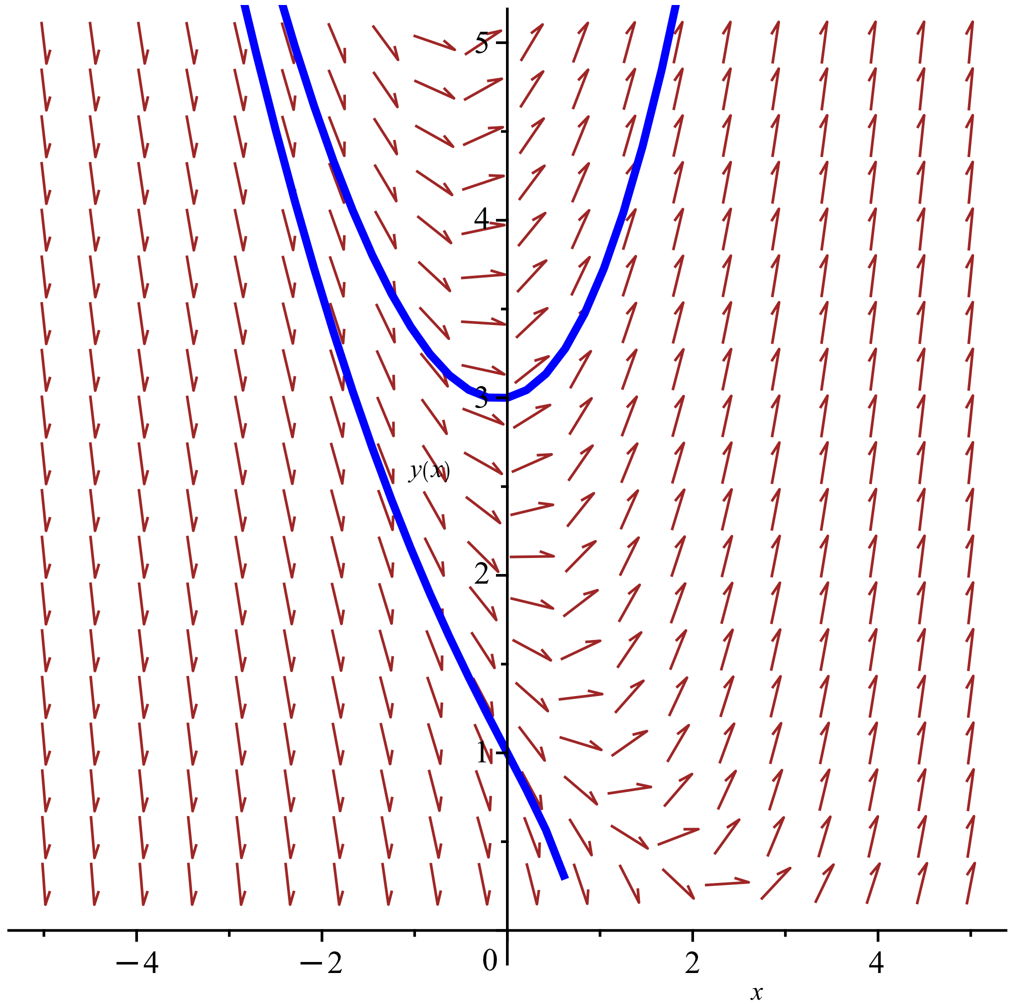
> $DEplot(ODE, y(x), x=-5..5, y=0..5)$



> $plot1 := DEplot(ODE, y(x), x = -5..5, y = 0..5, [y(0) = 1, y(0) = 3], linecolor = blue)$

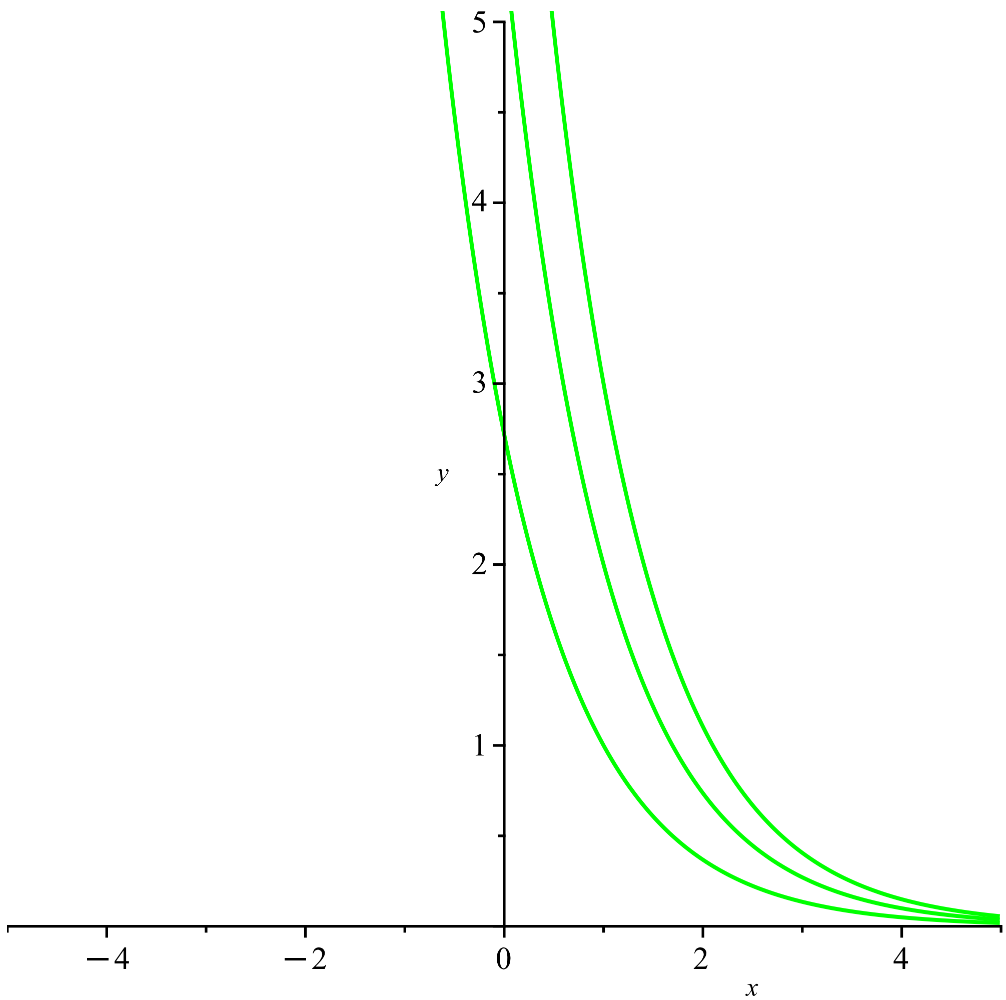
Будуємо інтегральні криві за початковою умовою :

```
Warning, plot may be incomplete, the following errors(s) were issued:
   cannot evaluate the solution further right of .75011235, accuracy
goal cannot be achieved with specified 'minstep'
```



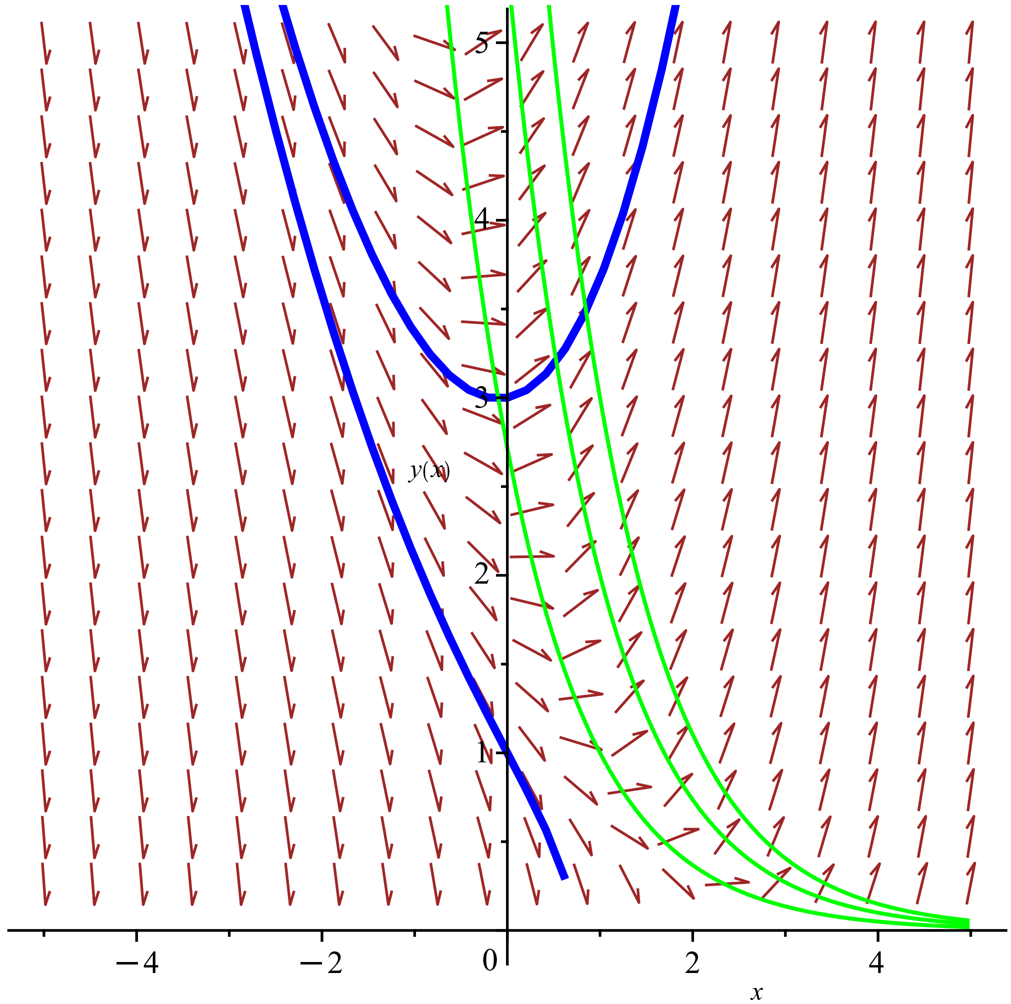
> $plot2 := plot([1 \cdot e^{(1-x)}, 2 \cdot e^{(1-x)}, 3 \cdot e^{(1-x)},], x = -5..5, y = 0..5, color = green)$

Ізоклін : $y = C \cdot e^{(1-x)}$



> $with(plots)$:;

> $display(plot1, plot2)$



> *restart*

в) $y' = arctgy + x,$ (0; 0), (0; -2);
Ізоклін: arctan(y(x)) = C - x
y(x) = tan(C - x)

> $ODE := diff(y(x), x) = \arctan(y(x)) - x$

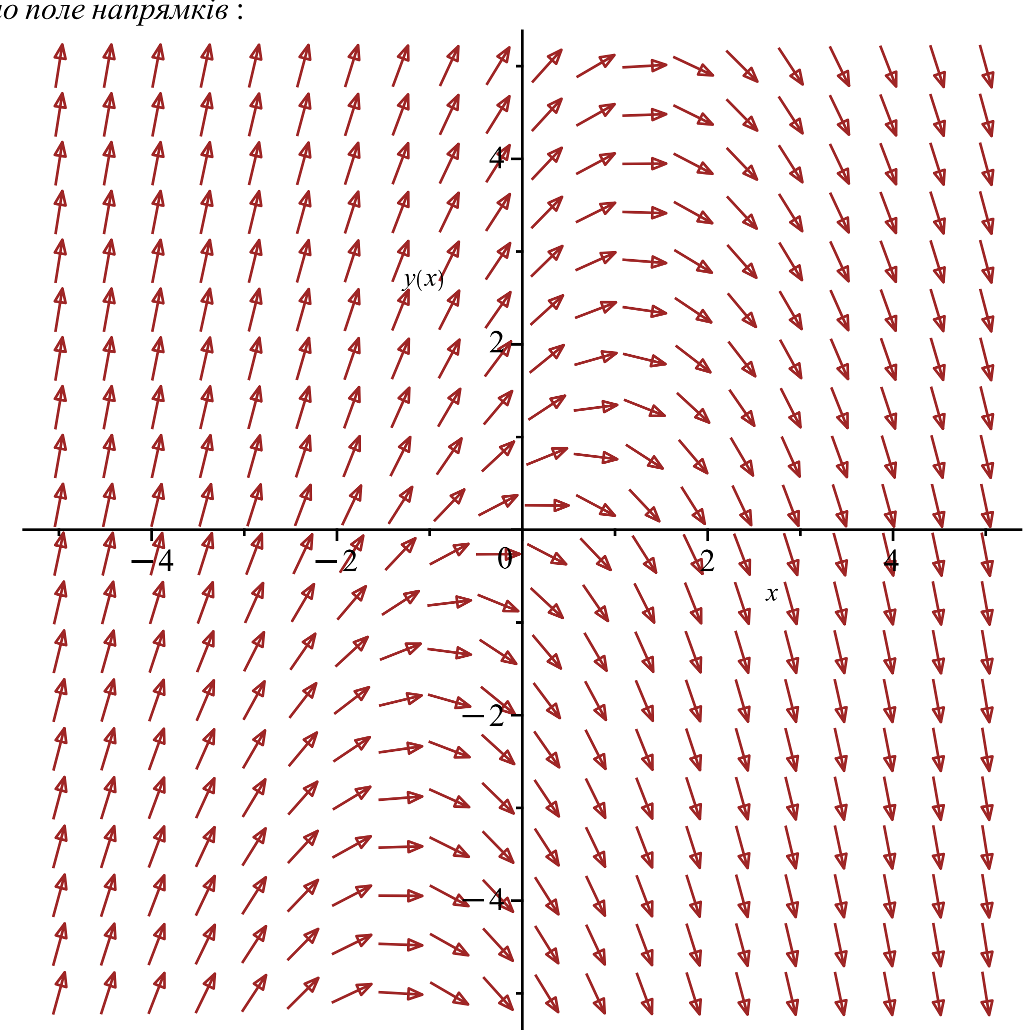
$$ODE := \frac{d}{dx} y(x) = \arctan(y(x)) - x \quad \textbf{(8)}$$

> *with*(*DEtools*)

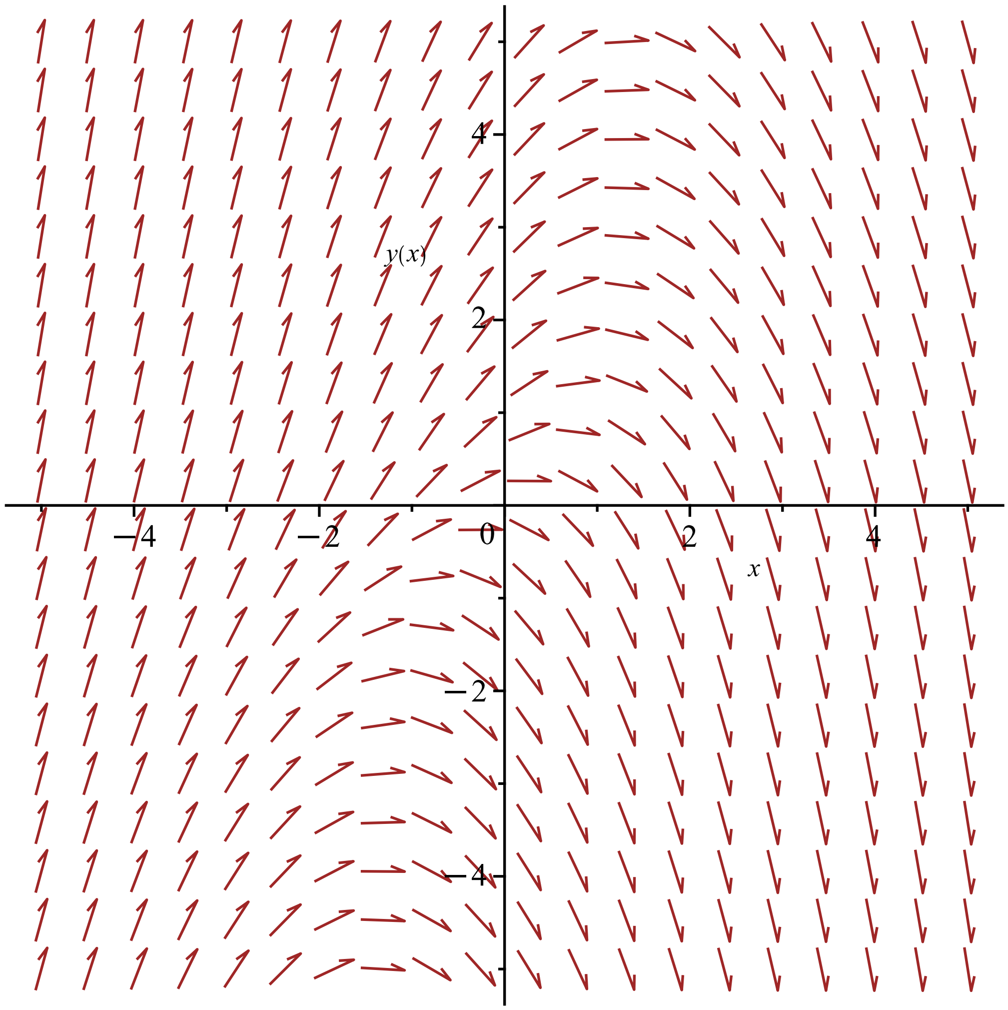
[*AreSimilar*, *Closure*, *DEnormal*, *DEplot*, *DEplot3d*, *DEplot_polygon*, *DFactor*, *DFactorLCLM*, *DFactorsols*, *Dchangevar*, *Desingularize*, *FindODE*, *FunctionDecomposition*, *GCRD*, *Gosper*, *Heunsols*, *Homomorphisms*, *IVPsol*, *IsHyperexponential*, *LCLM*, *MeijerGsols*, *MultiplicativeDecomposition*, *ODEInvariants*, *PDEchangecoords*, *PolynomialNormalForm*, *RationalCanonicalForm*, *ReduceHyperexp*, *RiemannPsols*, *Xchange*, *Xcommutator*, *Xgauge*, *Zeilberger*, *abelsol*, *adjoint*, *autonomous*, *bernoullisol*, *buildsol*, *buildsym*, *canoni*, *caseplot*, **(9)** 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]

> $dfieldplot(ODE, y(x), x=-5..5, y=-5..5, arrows=SLIM)$

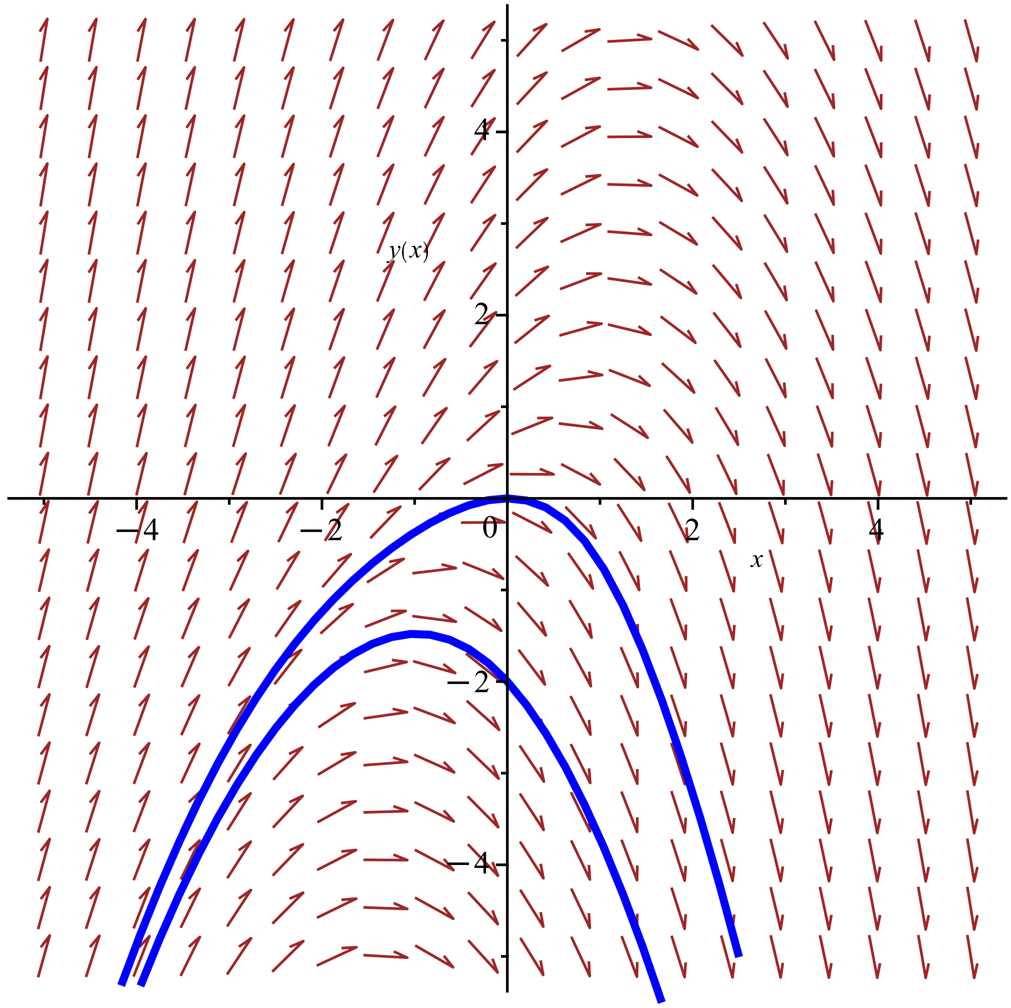
Будуємо поле напрямків :



> $DEplot(ODE, y(x), x=-5..5, y=-5..5)$

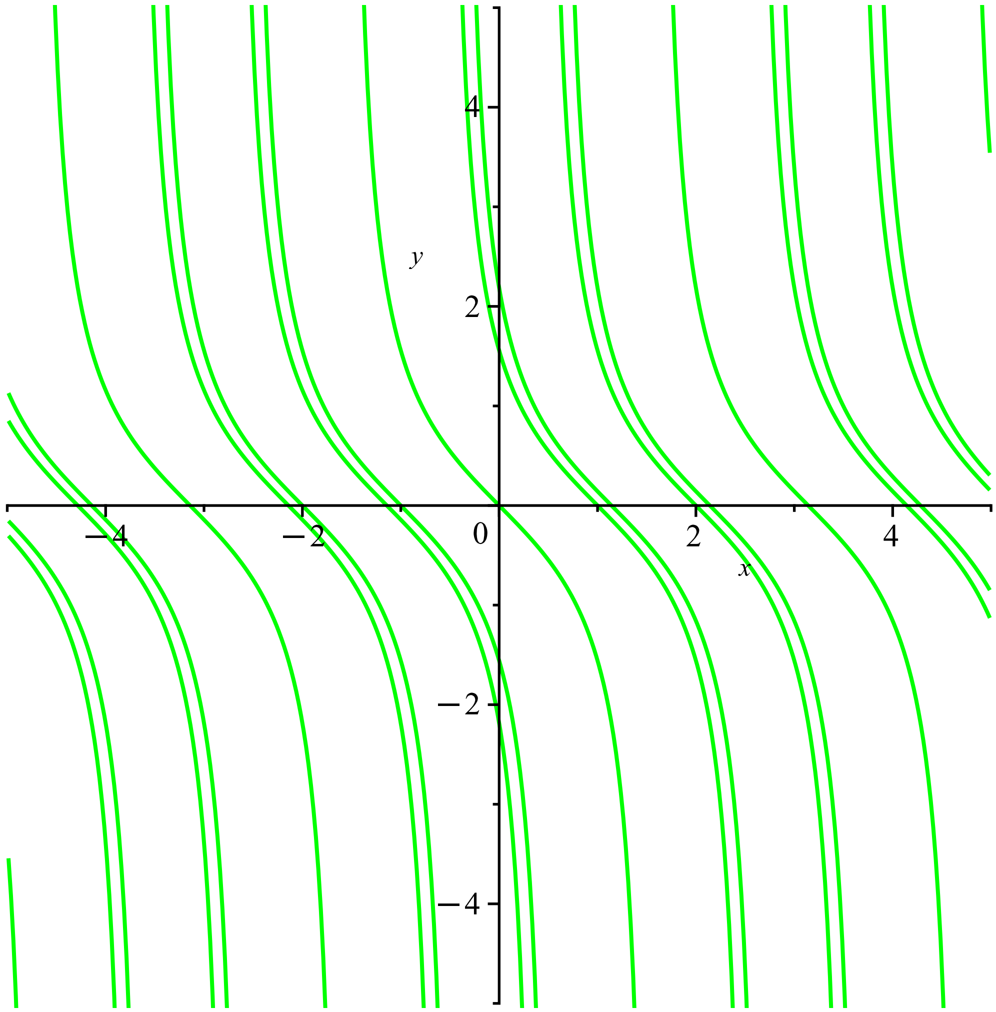


> $plot1 := DEplot(ODE, y(x), x = -5\,..5, y = -5\,..5\,, [\,y(0) = 0, y(0) = -2\,],$
$linecolor = blue)$



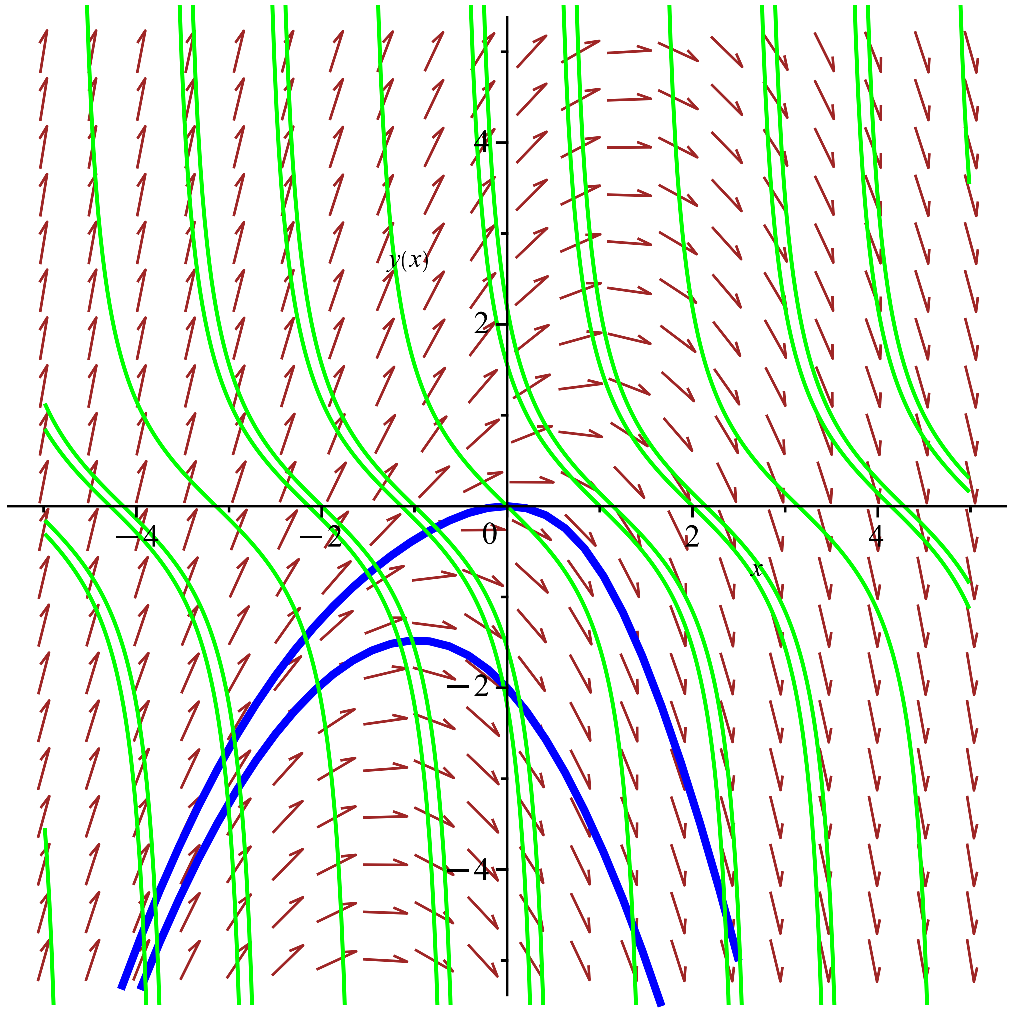
> $plot2 := plot([\tan(-2-x), \tan(-1-x), \tan(0-x), \tan(1-x), \tan(2-x)],$
> $x=-5..5, y=-5..5, color=green)$

Будуємо ізоклін



> $with(plots)$:;

> $display(plot1, plot2)$



> *restart*

г) $y' = x^2 + (y-1)^2$, (0; 0), (0; 3).

Ізоклін

$(y-1)^2 = \mathrm{C} - x^2$

$y - 1 = \pm\sqrt{C - x^2}$

$y = 1 \pm \sqrt{C - x^2}$

> $ODE := diff(y(x), x) = x^2 + (y-1)^2$

$$ODE := \frac{\mathrm{d}}{\mathrm{d}x}\, y(x) = x^2 + (y-1)^2 \quad \textbf{(10)}$$

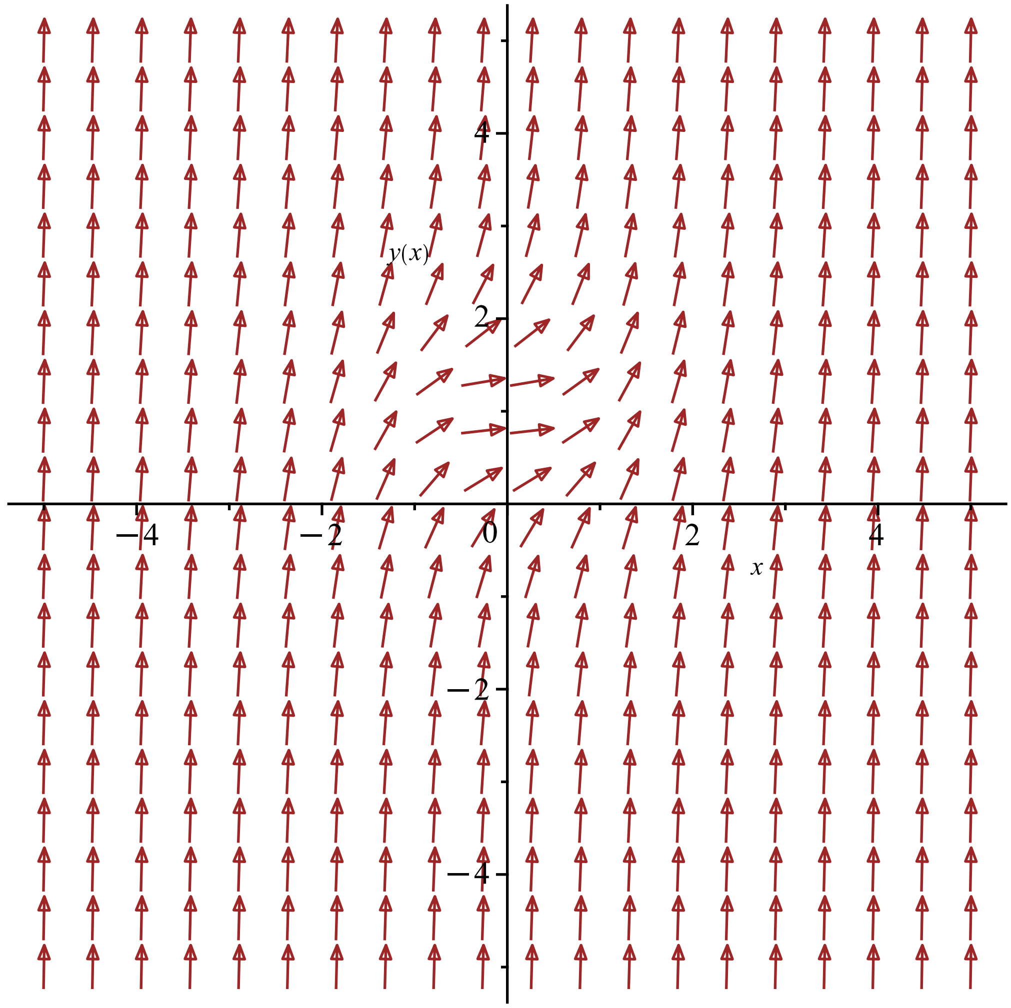
> *with*(*DEtools*)

Будуємо поле напрямків :

[*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*, **(11)** *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*]

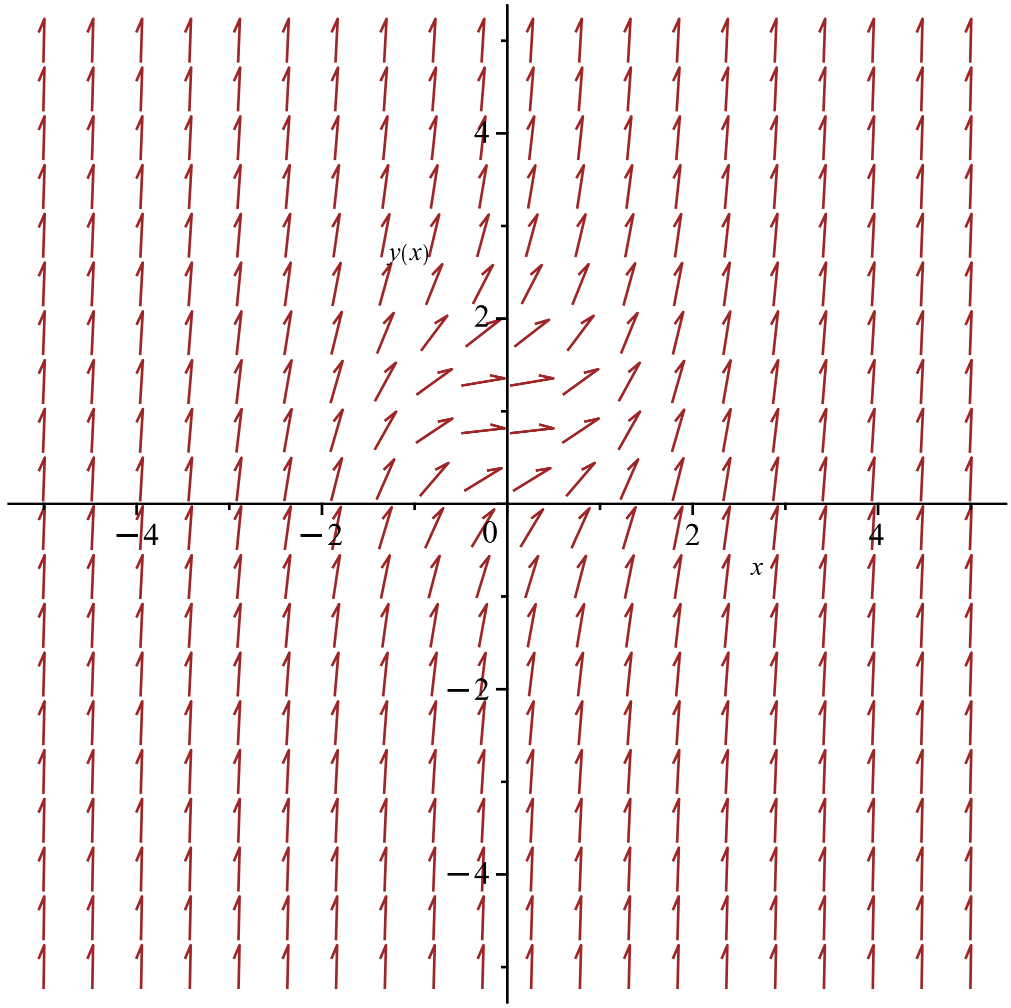
> $dfieldplot(ODE, y(x), x=-5..5, y=-5..5, arrows=SLIM)$

```
Warning, y is present as both a dependent variable and a name.
Inconsistent specification of the dependent variable is deprecated,
and it is assumed that the name is being used in place of the
dependent variable.
```



> $DEplot(ODE, y(x), x = -5 ..5, y = -5 ..5)$

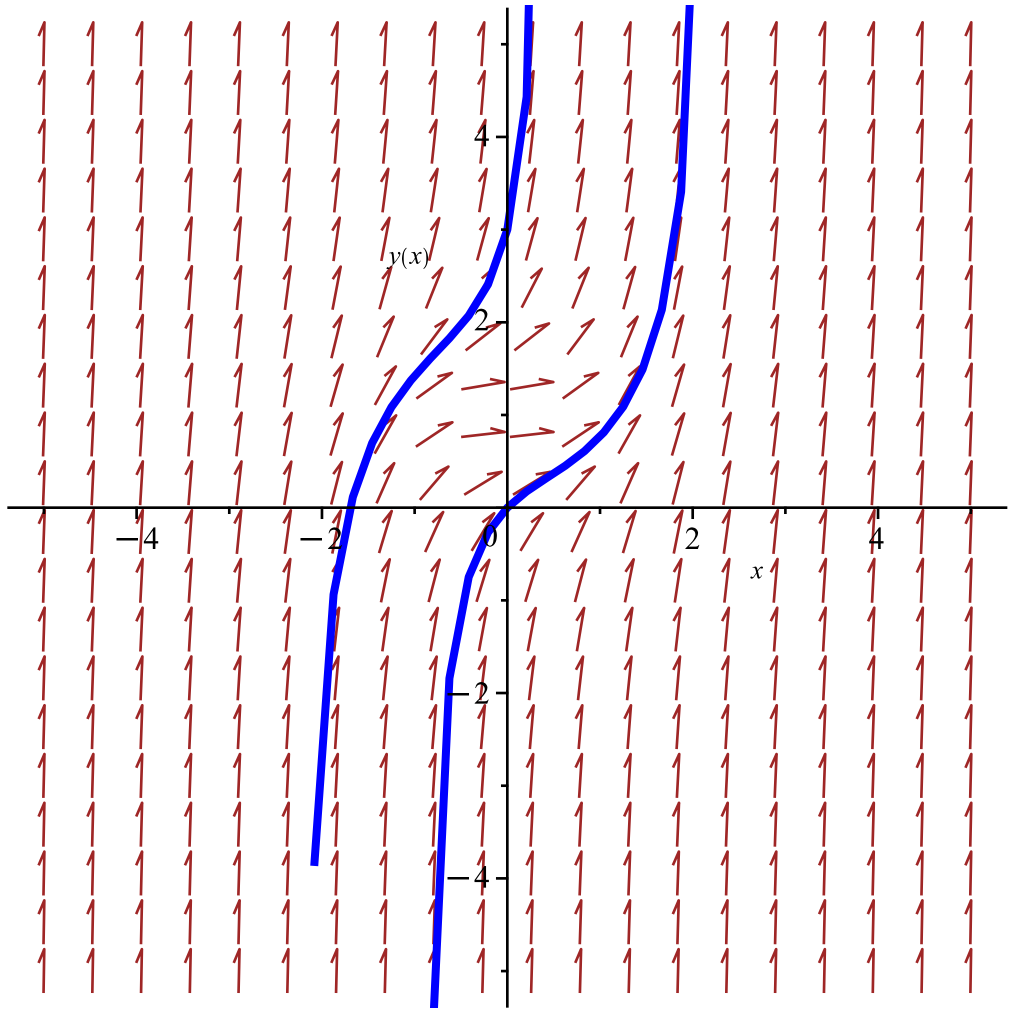
Warning, y is present as both a dependent variable and a name. Inconsistent specification of the dependent variable is deprecated, and it is assumed that the name is being used in place of the dependent variable.



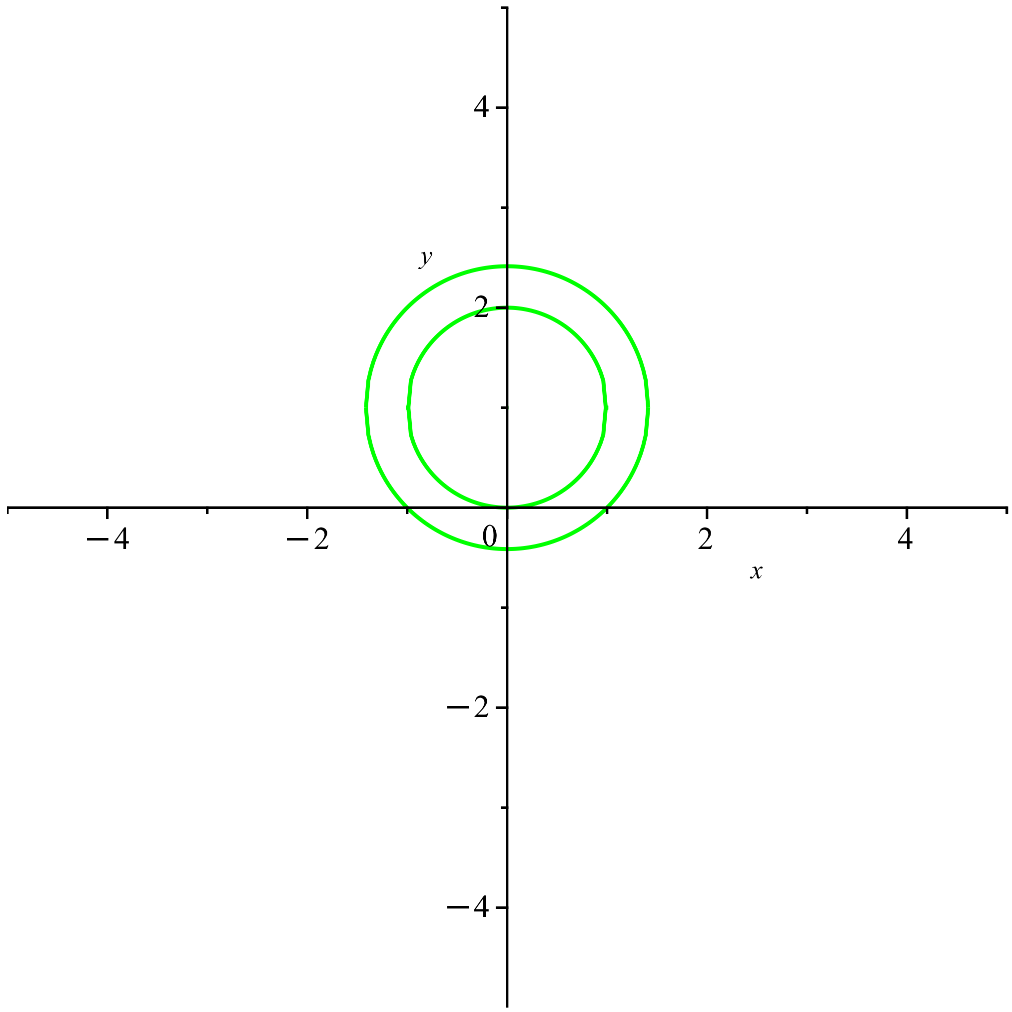
> $plot1 := DEplot(ODE, y(x), x = -5\,..5, y = -5\,..5\,, [y(0) = 0, y(0) = 3], linecolor = blue)$

Будуємо інтегральні криві за початковою умовою :

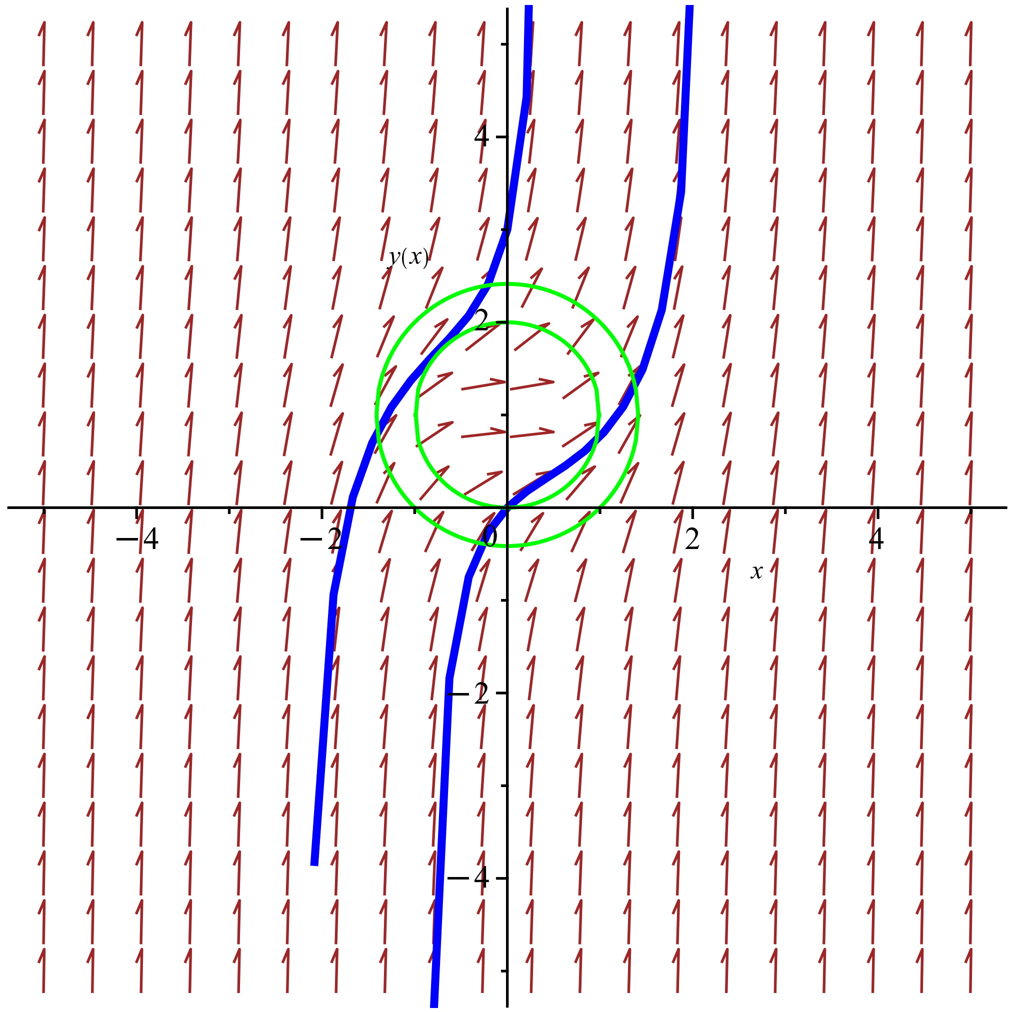
Warning, y is present as both a dependent variable and a name. Inconsistent specification of the dependent variable is deprecated, and it is assumed that the name is being used in place of the dependent variable.
Warning, plot may be incomplete, the following errors(s) were issued:
   cannot evaluate the solution further left of -2.2770340, probably a singularity



> $plot2 := plot\left(\left[1-\sqrt{-2-x^2}, 1-\sqrt{-1-x^2}, 1-\sqrt{-x^2}, 1-\sqrt{1-x^2}, 1-\sqrt{2-x^2}, 1+\sqrt{-2-x^2}, 1+\sqrt{-1-x^2}, 1+\sqrt{-x^2}, 1+\sqrt{1-x^2}, 1+\sqrt{2-x^2}\right], x=-5 ..5, y=-5 ..5, color=green\right)$




```
> with(plots) :;
> display(plot1, plot2)
```



> *restart*

Завдання виконано.

>

Задача 4. Методом ізоклін побудувати декілька інтегральних кривих диференціальних рівнянь

Варіант 14

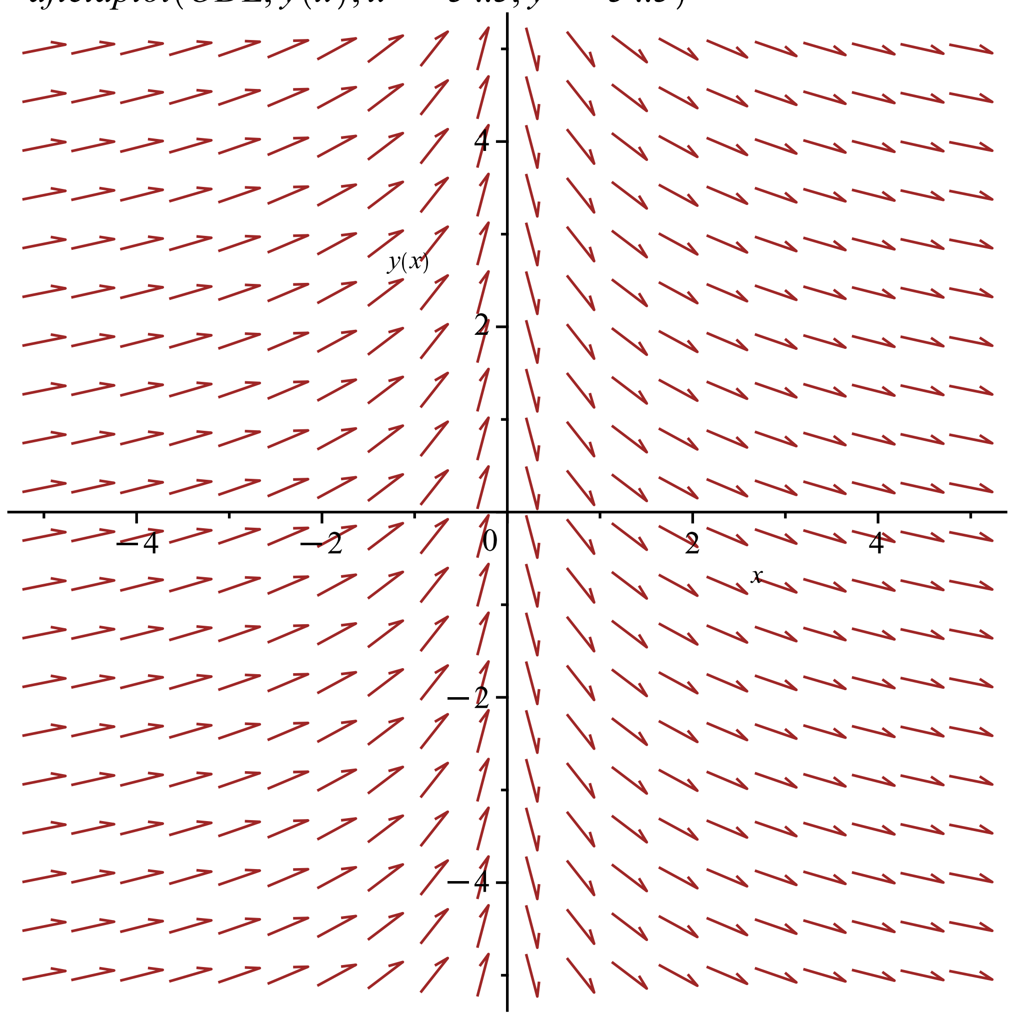
14) $xy' + 1 = 0$

> $ODE := x \cdot diff(y(x), x) + 1 = 0$

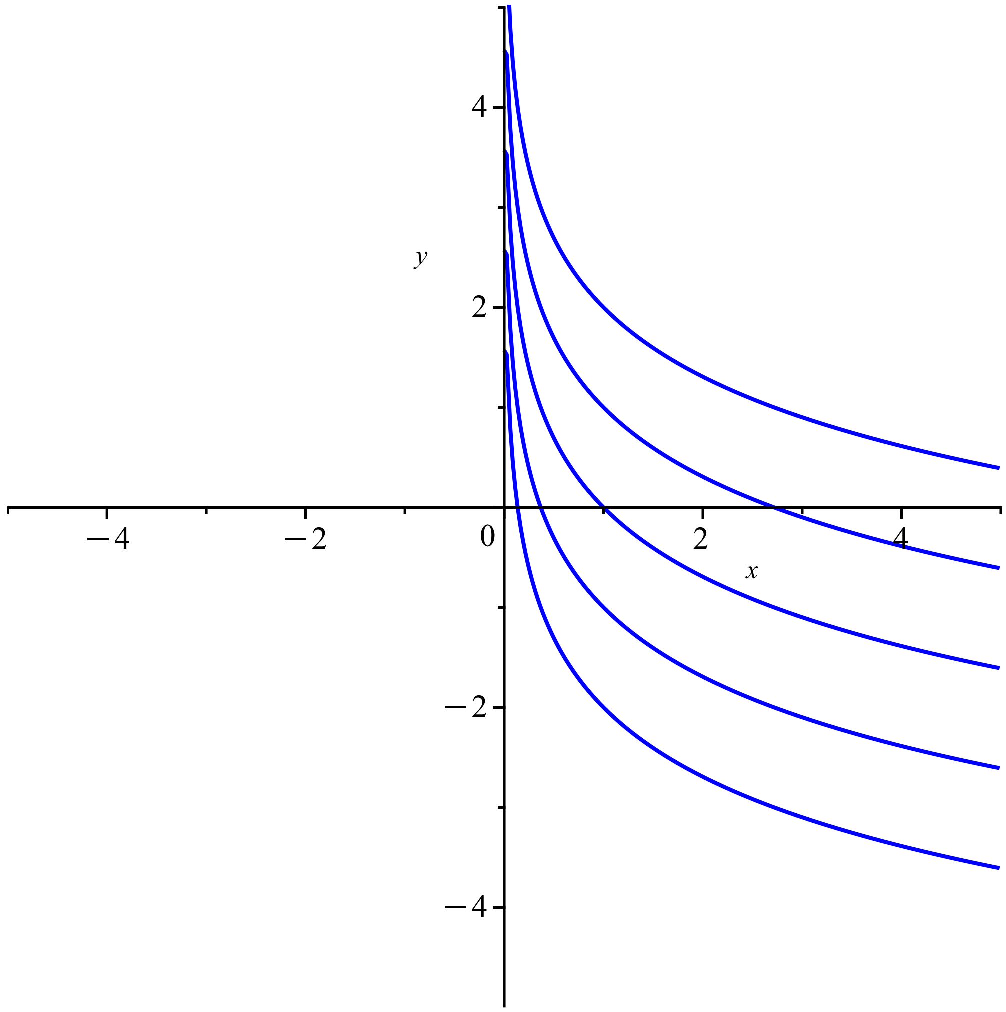
$$ODE := x \left(\frac{d}{dx} y(x) \right) + 1 = 0 \quad \textbf{(12)}$$

> $with(DETools):$

> $plot1 := dfieldplot(ODE, y(x), x = -5 ..5, y = -5 ..5)$

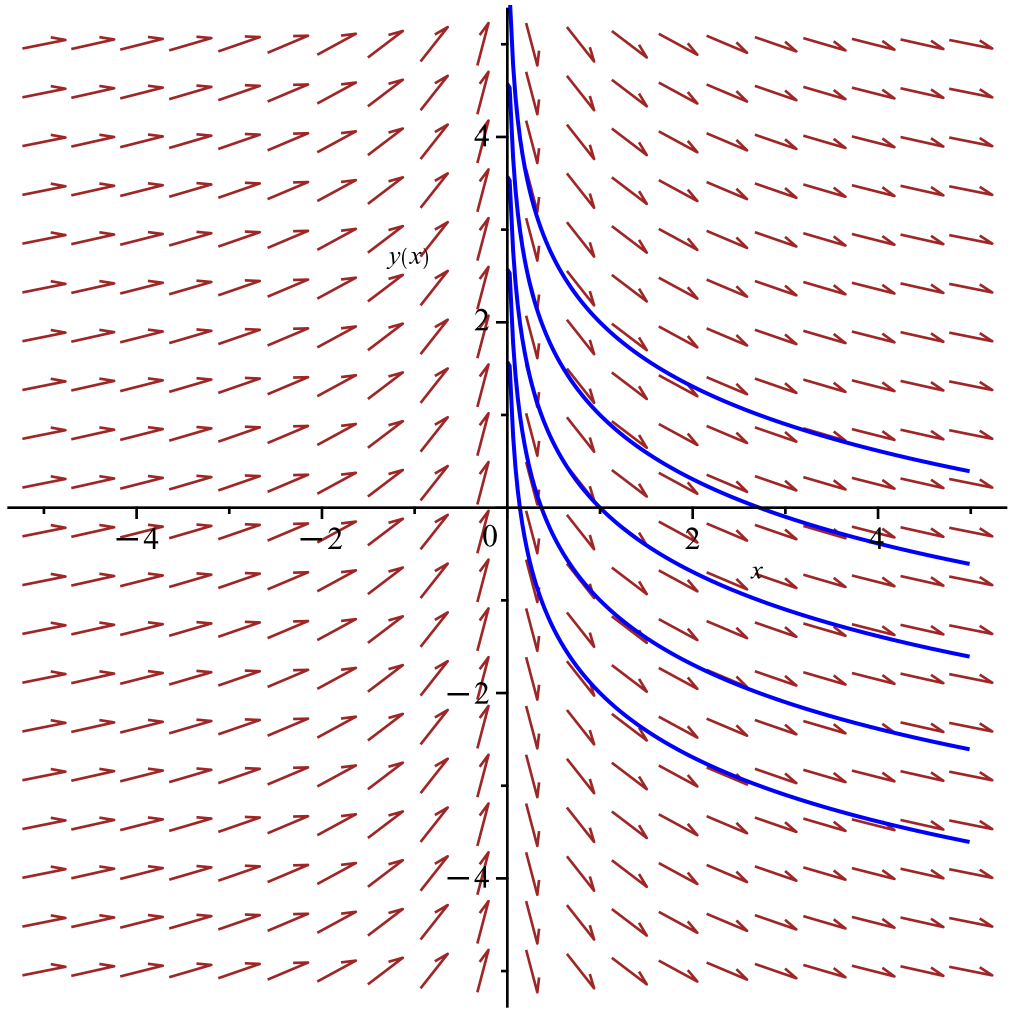


> $plot2 := plot([-2 - \ln(x), -1 - \ln(x), -\ln(x), 1 - \ln(x), 2 - \ln(x)], x = -5 ..5, y = -5 ..5, color = blue)$



> $with(plots):$

> $display(plot1, plot2)$



>

Завдання виконано.

Висновок **:**

В *результаті практичної роботи опанував навички побудови інтегральних кривих за допомогою ізоклін.*