*Київський національний університет*

*імені Тараса Шевченка*

*Факультет комп’ютерних наук та кібернетики*

*Лабораторна робота №1*

*з курсу*

*«Управління динамічними системами»*

*на тему:*

***«Аналітичне розв’язування диференціальних рівнянь***

***за допомогою комп’ютерних пакетів програм»***

*Виконала:*

*студентка групи ІПС-22*

*факультету комп’ютерних наук та кібернетики*

*Челушкіна Валерія*

***Київ-2024***

***ЗМІСТ.***

[***Умови задач.*** 3](#_Toc180621598)

[***Аналітичний розв’язок задач у зошиті.*** 4](#_Toc180621599)

[***Коди програм.*** 7](#_Toc180621600)

[***Результати робот програм.*** 13](#_Toc180621601)

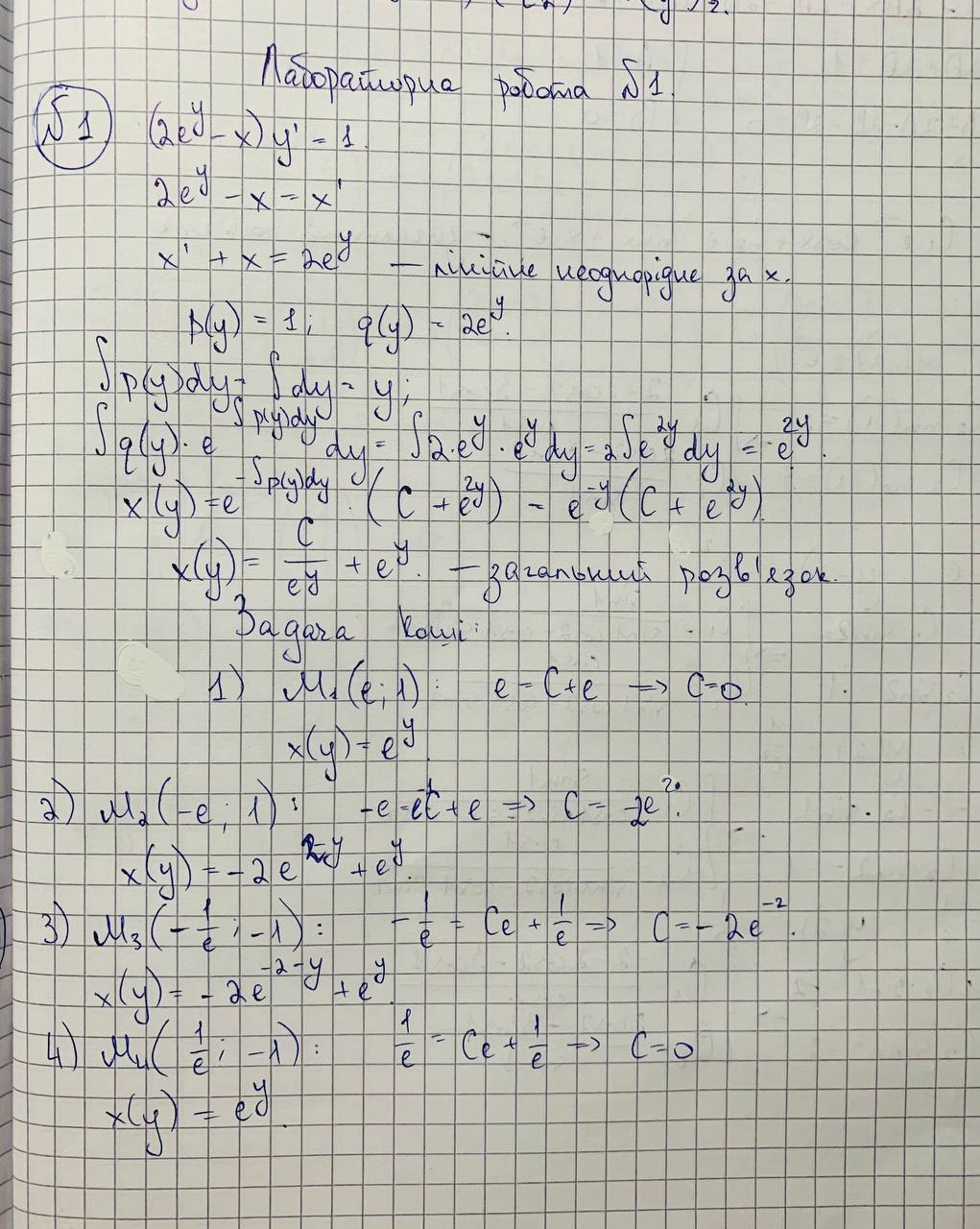
***Умови задач.***

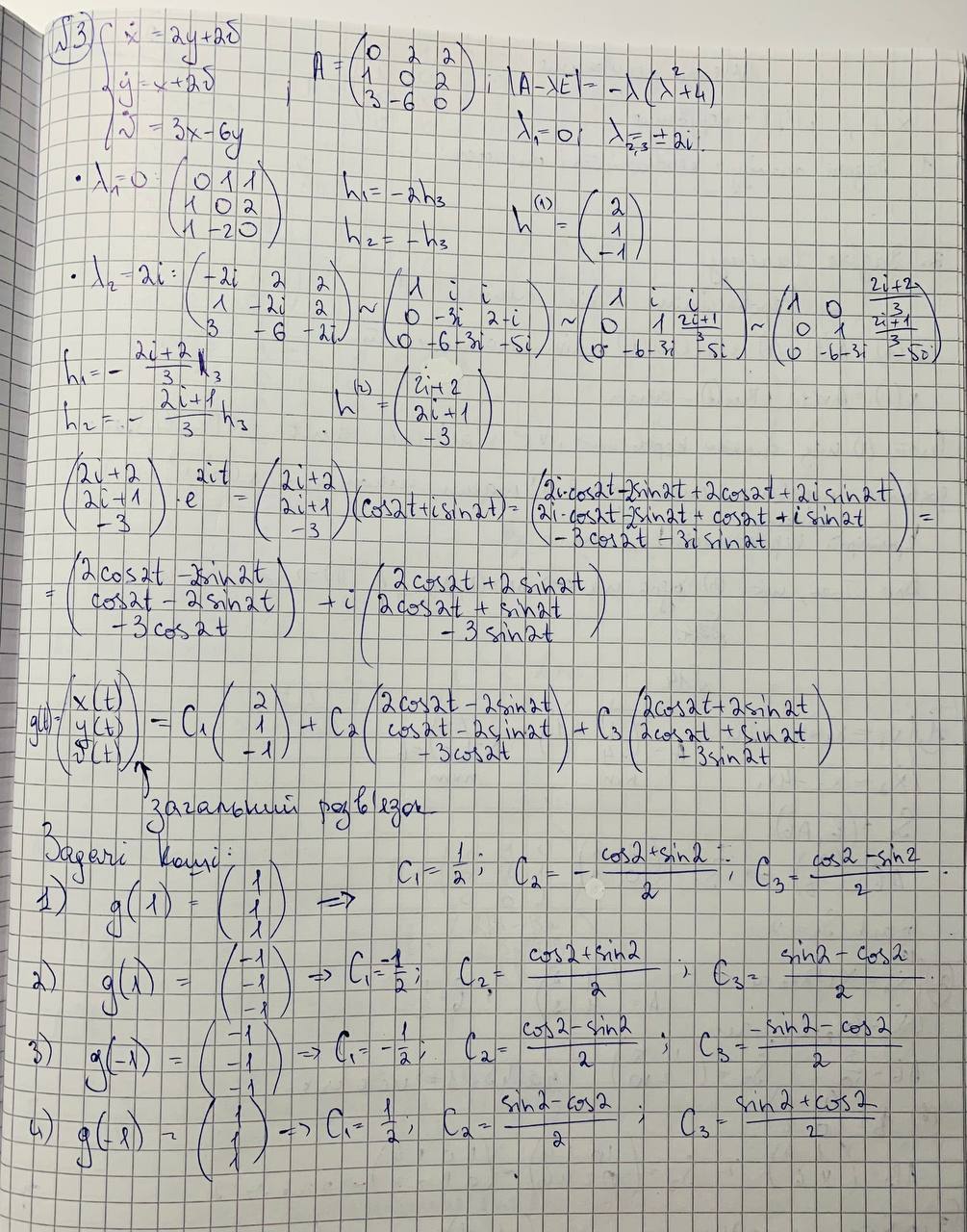
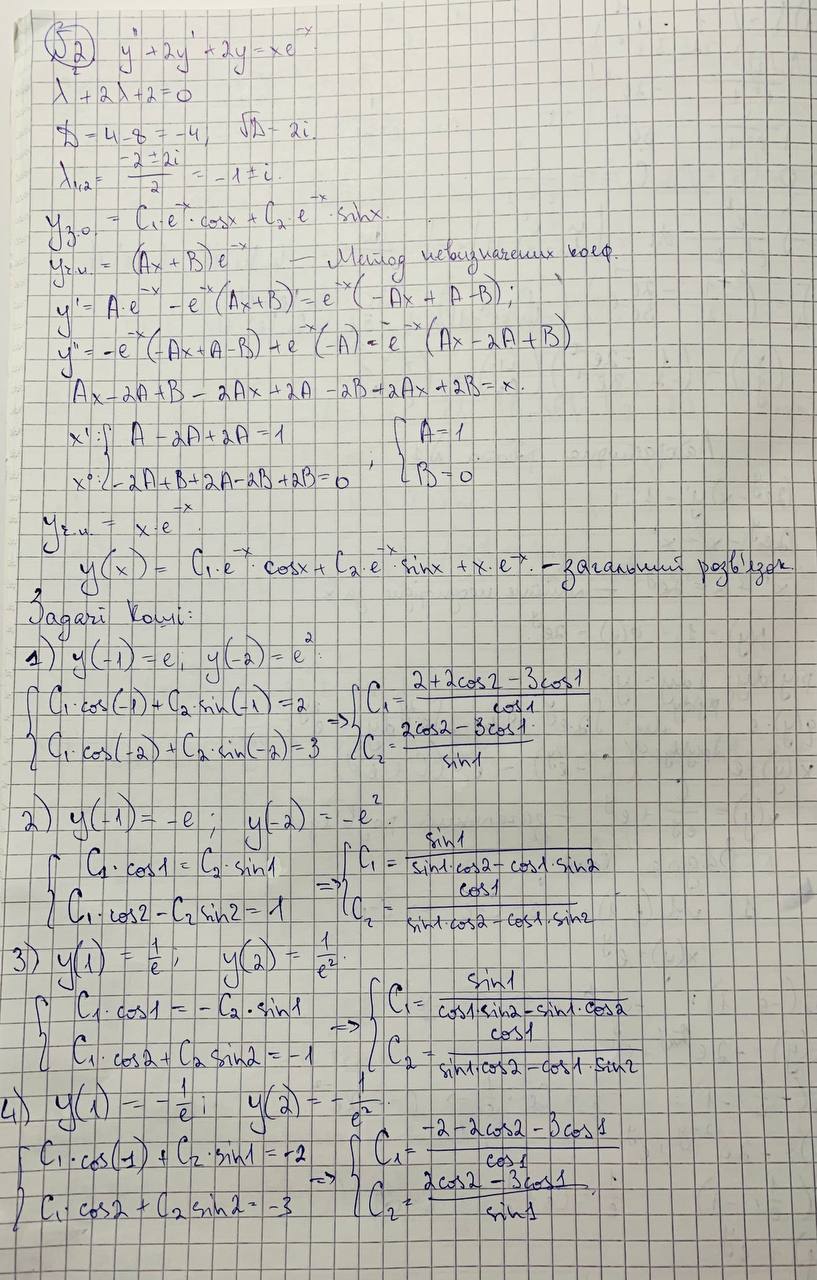
***№1.***

***№2.***

***№3.***

***Аналітичний розв’язок задач у зошиті.***





***Коди програм.***

***№1.***

***SageMath:***

*#general solution*

*y = function('y')(x)*

*de = diff(y, x) == 1 / (2\*exp(y)-x)*

*solution = desolve(de, y)*

*solution.show()*

*#Couchi problem solution*

*# (e, 1)*

*solution = desolve(de, y, ics = [e, 1])*

*solution.show()*

*# (-e,1)*

*solution = desolve(de, y, ics = [-e, 1])*

*solution.show()*

*# (-1/e, -1)*

*solution = desolve(de, y, ics = [-1/e, -1])*

*solution.show()*

*# (1/e, -1)*

*solution = desolve(de, y, ics = [1/e, -1])*

*solution.show()*

*#direction fields*

*x, y = var('x, y')*

*f(x, y) = 1 / (2 \* exp(y) - x)*

*p = plot\_slope\_field(f, (x, -5, 5), (y, -5, 5), headaxislength=3, headlength=3, axes\_labels=['$x$', '$y(x)$'])*

*e=2.71828182846*

*p += desolve\_rk4(f, y, ics=[e, 1], ivar=x, output = 'plot',*

*end\_points =[-5, 5], thickness=2, rgbcolor=hue(1))*

*p += desolve\_rk4(f, y, ics=[-e,1], ivar=x, output = 'plot',*

*end\_points =[-5, 5], thickness=2, rgbcolor=hue(0.2))*

*p += desolve\_rk4(f, y, ics=[-1/e, -1], ivar=x, output = 'plot',*

*end\_points =[-5, 5], thickness=2, rgbcolor=hue(0.4))*

*p += desolve\_rk4(f, y, ics=[1/e, -1], ivar=x, output = 'plot',*

*end\_points =[-5, 5], thickness=2, rgbcolor=hue(0.8))*

*show(p, xmin=-5, xmax=5, ymin=-5, ymax=5)*

***Wolfram Mathematica:***

*eqn = (2\*E^(y[x]) - x)\*y'[x] == 1*

*generalsolution = DSolve[eqn, y, x]*

*couch1 = DSolve[{eqn, y[E] == 1}, y, x]*

*couch2 = DSolve[{eqn, y[-E] == 1}, y, x]*

*couch3 = DSolve[{eqn, y[1/E] == -1}, y, x]*

*couch4 = DSolve[{eqn, y[-1/E] == -1}, y, x]*

*vplot = VectorPlot[{1, eqn}, {x, -5, 5}, {y, -5, 5},*

*VectorStyle -> {Thick, Red}, VectorScaling -> {.03, .03, None},*

*VectorPoints -> 25];*

*couchiplot1 =*

*Plot[y[x] /. couch1, {x, -5, 5}, PlotStyle -> {Thick, Red}];*

*couchiplot2 =*

*Plot[y[x] /. couch2, {x, -5, 5}, PlotStyle -> {Thick, Yellow}];*

*couchiplot3 =*

*Plot[y[x] /. couch3, {x, -5, 5}, PlotStyle -> {Thick, Purple}];*

*couchiplot4 =*

*Plot[y[x] /. couch4, {x, -5, 5}, PlotStyle -> {Thick, Green}];*

*Show[vplot, couchiplot1, couchiplot2, couchiplot3, couchiplot4 ,*

*Frame -> False, Axes -> True, AxesLabel -> {"x", "y"}]*

***№2.***

***SageMath:***

*#general solution*

*show("General solution:")*

*y = function('y')(x)*

*de =diff(diff(y,x),x) + 2 \* diff(y,x) + 2\*y - x\*e^(-x)*

*solution = desolve(de, y)*

*solution.show()*

*#Couchi problem solution*

*# (-1,e,-2,e^2)*

*show("y(-1)=e; y(-2)=e^2:")*

*solution=desolve(de,y,ics=[-1,e,-2,e^2])*

*solution.show()*

*#(-1,-e,-2,-e^2)*

*show("y(-1)=-e; y(-2)=-e^2:")*

*solution1=desolve(de,y,ics=[-1,-e,-2,-e^2])*

*solution1.show()*

*#(1,1/e,2,1/e^(2))*

*show("y(1)=1/e; y(2)=1/e^2:")*

*solution2=desolve(de,y,ics=[1,1/e,2,1/e^(2)])*

*solution2.show()*

*#(1,-1/e,2,-1/e^(2))*

*show("y(1)=-1/e; y(2)=-1/e^2:")*

*solution3=desolve(de,y,ics=[1,-1/e,2,-1/e^(2)])*

*solution3.show()*

***Wolfram Mathematica:***

*eqn = (y''[x] + 2\*y'[x] + 2\*y[x]) == x\*E^(-x)*

*generalsolution = DSolve[eqn, y, x]*

*couch1 = DSolve[{eqn, y[-1] == E, y[-2] == E^2}, y, x]*

*couch2 = DSolve[{eqn, y[-1] == -E, y[-2] == -E^2}, y, x]*

*couch3 = DSolve[{eqn, y[1] == 1/E, y[2] == 1/(E^2)}, y, x]*

*couch4 = DSolve[{eqn, y[1] == -1/E, y[2] == -1/(E^2)}, y, x]*

*vplot = VectorPlot[{1, eqn}, {x, -5, 5}, {y, -5, 5},*

*VectorStyle -> {Thick, Red}, VectorScaling -> {.03, .03, None},*

*VectorPoints -> 25];*

*couchiplot1 =*

*Plot[y[x] /. couch1, {x, -5, 5}, PlotStyle -> {Thick, Blue}];*

*couchiplot2 =*

*Plot[y[x] /. couch2, {x, -5, 5}, PlotStyle -> {Thick, Red}];*

*couchiplot3 =*

*Plot[y[x] /. couch3, {x, -5, 5}, PlotStyle -> {Thick, Green}];*

*couchiplot4 =*

*Plot[y[x] /. couch4, {x, -5, 5}, PlotStyle -> {Thick, Purple}];*

*Show[vplot, couchiplot1, couchiplot2, couchiplot3, couchiplot4,*

*Frame -> False, Axes -> True, AxesLabel -> {"x", "y"}]*

***№3.***

***SageMath:***

*#general solution*

*show("General solution:")*

*t = var('t')*

*x =function ('x')(t)*

*y =function ('y')(t)*

*v =function ('v')(t)*

*de1 = diff(x,t) == 2\*y+2\*v*

*de2 = diff(y,t) == x+2\*v*

*de3 = diff(v,t) == 3\*x-6\*y*

*sol = desolve\_system([de1,de2,de3],[x,y,v],ivar=t)*

*solx,soly,solv = sol[0].rhs(),sol[1].rhs(),sol[2].rhs()*

*solution = matrix ([[solx],[soly],[solv]])*

*show(solution)*

*#Couchi problem solution*

*#(1, 1, 1, 1)*

*show("x=1, y=1, v=1, t=1:")*

*sol = desolve\_system([de1, de2, de3], [x, y, v], ivar=t, ics=[1, 1, 1, 1])*

*solx, soly, solv = sol[0].rhs(), sol[1].rhs(), sol[2].rhs()*

*solution = matrix([[solx], [soly], [solv]])*

*show(solution)*

*#(-1, -1, -1, 1)*

*show("x=-1, y=-1, v=-1, t=1:")*

*sol = desolve\_system([de1, de2, de3], [x, y, v], ivar=t, ics=[-1, -1, -1, 1])*

*solx, soly, solv = sol[0].rhs(), sol[1].rhs(), sol[2].rhs()*

*solution = matrix([[solx], [soly], [solv]])*

*show(solution)*

*#(-1, -1, -1, -1)*

*show("x=-1, y=-1, v=-1, t=-1:")*

*sol = desolve\_system([de1, de2, de3], [x, y, v], ivar=t, ics=[-1, -1, -1, -1])*

*solx, soly, solv = sol[0].rhs(), sol[1].rhs(), sol[2].rhs()*

*solution = matrix([[solx], [soly], [solv]])*

*show(solution)*

*#(1, 1, 1, -1)*

*show("x=1, y=1, v=1, t=-1:")*

*sol = desolve\_system([de1, de2, de3], [x, y, v], ivar=t, ics=[1, 1, 1, -1])*

*solx, soly, solv = sol[0].rhs(), sol[1].rhs(), sol[2].rhs()*

*solution = matrix([[solx], [soly], [solv]])*

*show(solution)*

*x, y, v = var('x y v')*

*dx = diff(x,t) == 2\*y+2\*v*

*dy = diff(y,t) == x+2\*v*

*dv = diff(v,t) == 3\*x-6\*y*

*show("Direction field:")*

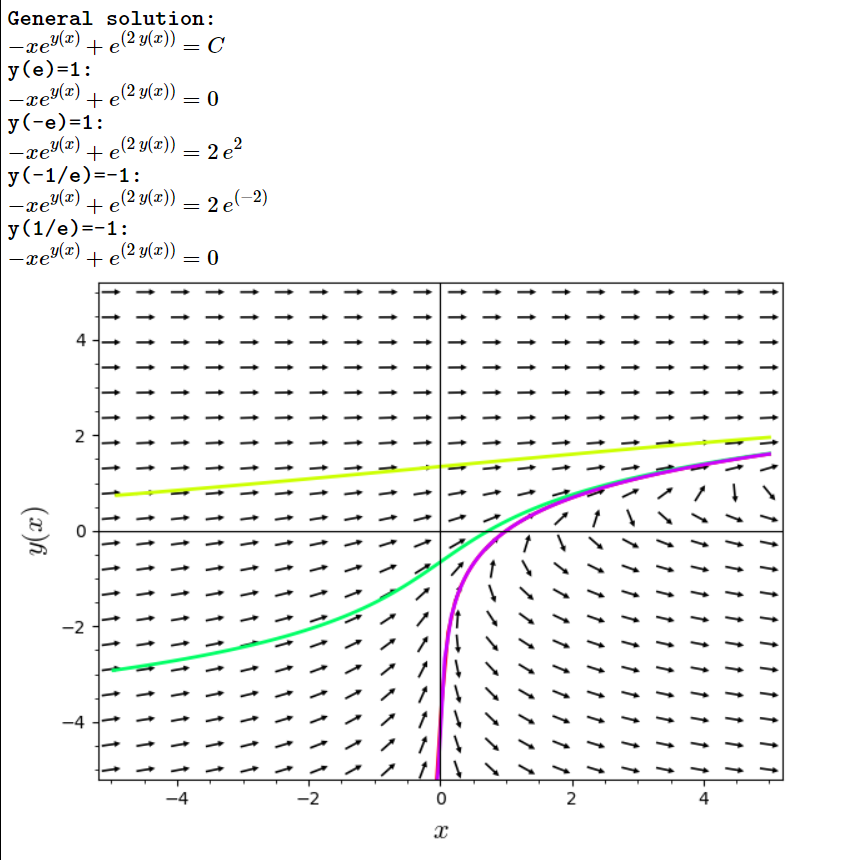
*p = plot\_vector\_field3d((dx, dy, dv), (x, -1, 1), (y, -1, 1), (v, -1, 1))*

*p.show(xmin=-10, xmax=10, ymin=-10, ymax=10, vmin=-10, vmax=10)*

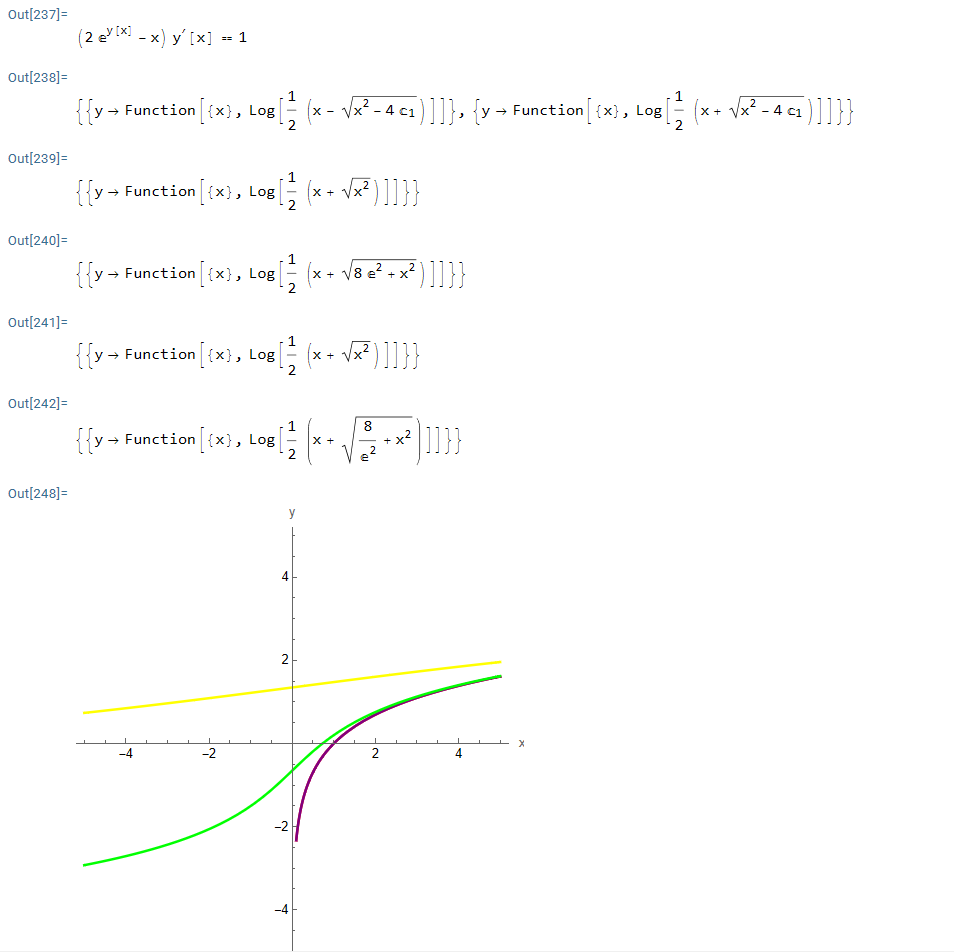
***Результати робот програм.***

***№1.***

***SageMath:***

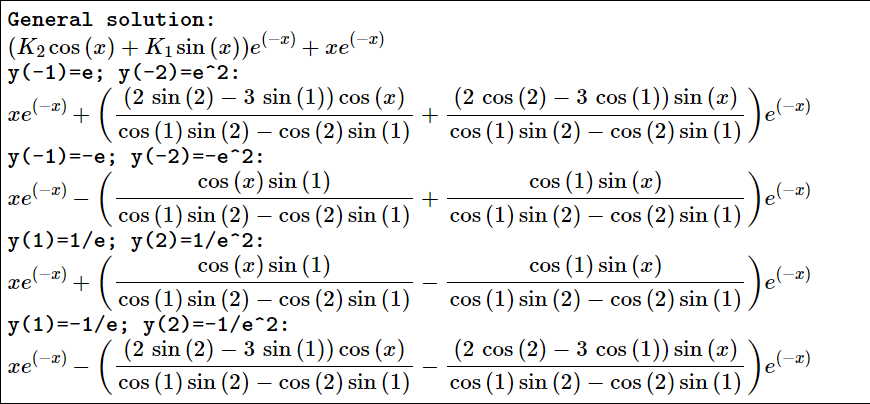
**

***Wolfram Mathematica:***

**

***№2.***

***SageMath:***

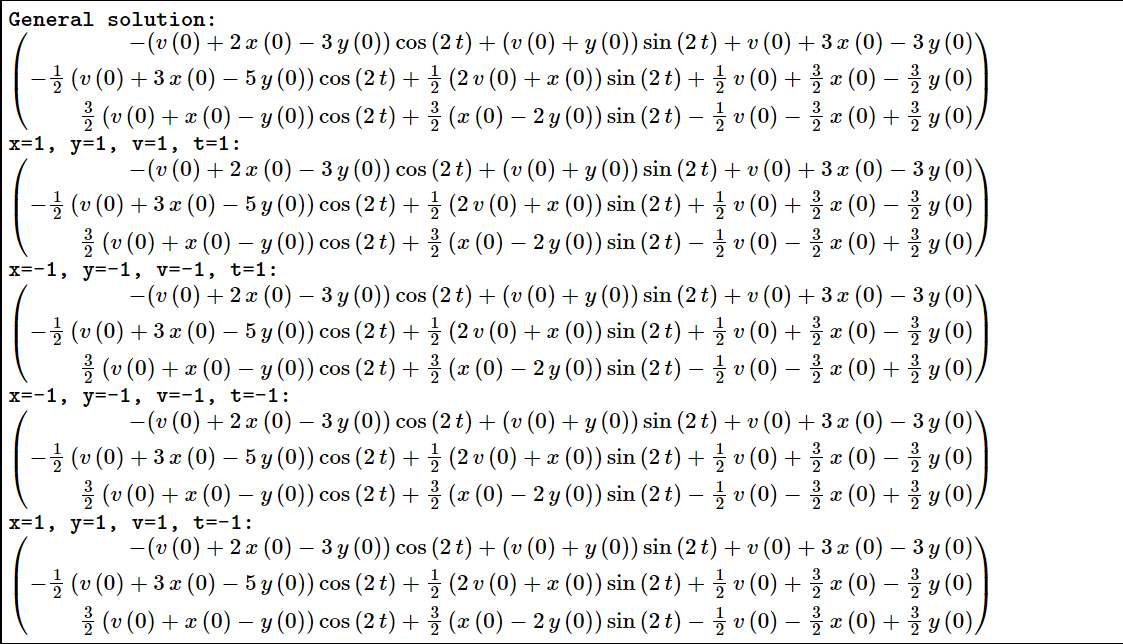
**

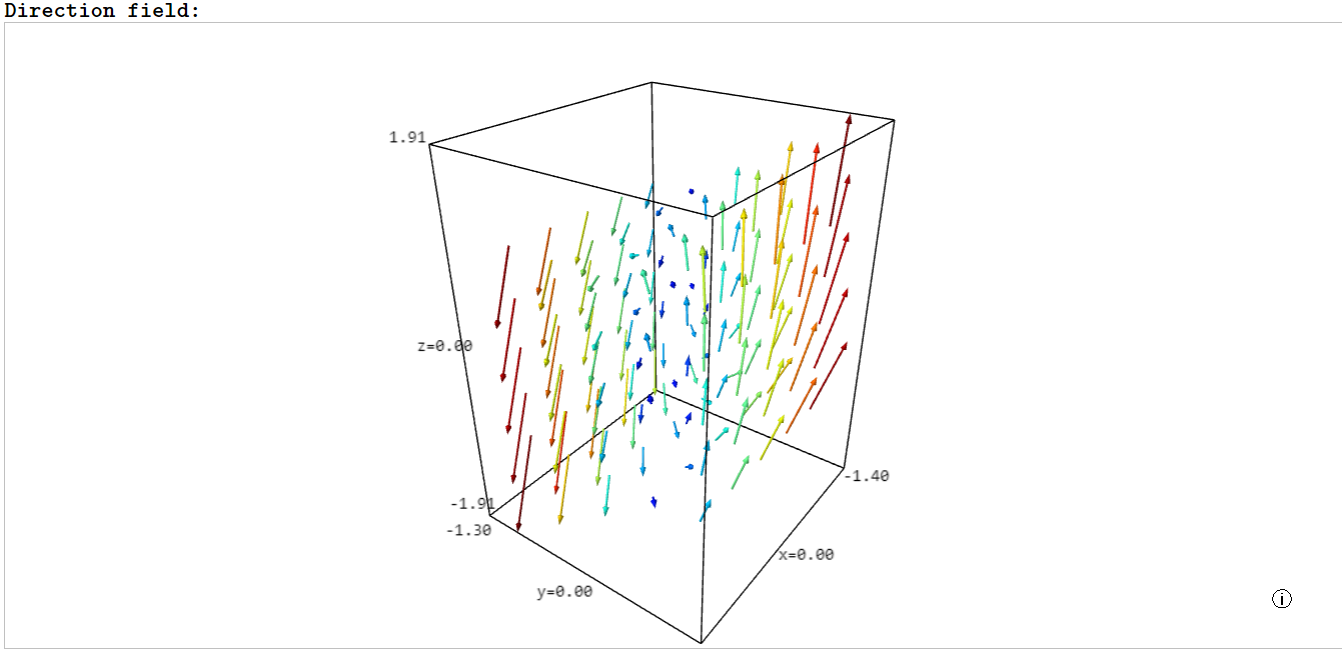
***Wolfram Mathematica:***

**

***№3.***

***SageMath:***

**

**