> **################################## P1 #####################################**

**restart;**

> **Digits:=15;**



> **eq:=diff(y(t),t)=-y(t);**



> **sol:=dsolve({eq,y(0)=1},type=numeric):**

> **sol(1);**



> **f:=rhs(eq);**



> **f:=subs(y(t)=Y,f);**



> **F:=unapply(f,Y);**



> **F(0);F(1);**





> **YY[0]:=1.0;**



> **h:=0.1;**



> **YY[1]:=YY[0]+h\*F(YY[0]);**



> **for i from 2 to 10 do YY[i]:=YY[i-1]+h\*F(YY[i-1]);od;**



















> **err:=abs(subs(sol(1),y(t))-YY[10]);**



> **Eulerforward:=proc(f,y0,tf,N)**

**local F,h,YY,i;**

**F:=unapply(f,Y);**

**h:=tf/N;**

**YY[0]:=y0;**

**for i from 1 to N do**

**YY[i]:=YY[i-1]+h\*F(YY[i-1]);**

**od;**

**#[seq([i\*h,YY[i]],i=0..N)];#for printing all the values**

**YY[N];**

**end proc;**



> **f;**



> **sol(1);**



> **sol3:=Eulerforward(f,1.0,1.0,2);**



> **for i from 1 to 10 do YY[i]:=Eulerforward(f,1.0,1.0,2^i);od;**





















> **for i from 1 to 9 do**

**Y1[i]:=abs(YY[i+1]-YY[i]);**

**od;**



















> **for i from 1 to 8 do Y1[i]/Y1[i+1];od;**

















> **#from last homework:**

**AutoEuler:=proc(f,y0,tf,N)**

**local F,h,yy,i;**

**F2:=unapply(f2,seq(Y[i],i=1..nops(y0)));**

**h:=tf/N;**

**if nops(y0)>1 then**

**#print("yes");**

**yy[0]:=y0;**

**for j from 1 to N do**

**yy[j]:=yy[j-1]+h\*F2(seq(yy[j-1][i],i=1..nops(y0)));**

**#Y1[j][1]:=abs(yy[j][1]-yy[j-1][1]);**

**#Y1[j][2]:=abs(yy[j][2]-yy[j-1][2]);**

**#Y11[j][1]:=Y1[j][1]/Y1[j-1][1];**

**#Y11[j][2]:=Y1[j][2]/Y1[j-1][2];**

**#print(Y11[j][1]);**

**#print(Y11[j][2]);**

**od;**

**#print(yy[N]);**

**end if;**

**if nops(y0)=1 then yy[0]:=y0; print("no");**

**for i from 1 to N do**

**yy[i]:=yy[i-1]+h\*F(yy[i-1]);**

**od;**

**#return yy[N];**

**end if;**

**#h;**

**yy[N];**

**end proc;**

Warning, `F2` is implicitly declared local to procedure `AutoEuler`

Warning, `j` is implicitly declared local to procedure `AutoEuler`



> **eq11:=diff(y[1](t),t)=-y[1](t)^2;**

**eq2:=diff(y[2](t),t)=y[1](t)^2-y[2](t);**

**f2:=subs([y[1](t)=Y[1],y[2](t)=Y[2]],[rhs(eq11),rhs(eq2)]);**







> **sol2:=dsolve({eq11,eq2,y[1](0)=1,y[2](0)=0},type=numeric):**

> **sol2(1);**



> **eff:=[1.0,0.0];**



>

**for i from 1 to 10 do YY[i]:=AutoEuler(f2,eff,1.0,2^i);od;**





















> **for i from 1 to 9 do**

**Y1[i][1]:=abs(YY[i+1][1]-YY[i][1]);**

**Y1[i][2]:=abs(YY[i+1][2]-YY[i][2]);**

**od;**

**for i from 1 to 9 do**

**[Y1[i][1],Y1[i][2]];**

**od;**























































> **for i from 1 to 8 do**

**[Y1[i][1]/Y1[i+1][1],Y1[i][2]/Y1[i+1][2]];**

**od;**

















> **################################## P2 #####################################**

> **restart;**

**NewtonRhapson:=proc(E,V,T,G,jac)**

**local f,F,JAC,xold,tol,xnew;**

**with(ArrayTools):**

**with(LinearAlgebra):**

**f:=E;**

**F:=unapply(f,x);**

**#jac:=VectorCalculus:-Jacobian(f,[seq(x[i],i=1..nops(V))]);**

**#print(jac);**

**Jac:=unapply(jac,x);**

**xold:=G;**

**tol:=1;**

**for i from 1 to 50 do**

**dx:=LinearAlgebra:-LinearSolve(-Jac(xold),F(xold));**

**#print(Jac(xold));**

**xnew:=xold+dx;**

**tol:=VectorCalculus[Norm](dx);**

**xold:=xnew;**

**if tol < T then break end if**

**end do;**

**end proc;**

Warning, `Jac` is implicitly declared local to procedure `NewtonRhapson`

Warning, `i` is implicitly declared local to procedure `NewtonRhapson`

Warning, `dx` is implicitly declared local to procedure `NewtonRhapson`



> **Digits:=15;**



> **eq1:=Y1=y(t-h);**



> **eq1:=Y1=series(y(t-h),h);**



> **eq1:=Y1=series(y(t-h),h,3);**



> **eq1:=y0=convert(series(y(t-h),h,2),polynom);**



> **eq1:=convert(eq1,diff);**



> **eq2:=subs(diff(y(t),t)=F1,y(t)=Y1,eq1);**



> **eq3:=Y1=solve(eq2,Y1);**



> **eq:=diff(y(t),t)=-y(t);**



> **sol:=dsolve({eq,y(0)=1},type=numeric):**

> **sol(1);**



> **f:=rhs(eq);**



> **f:=subs(y(t)=Y,f);**



> **F:=unapply(h\*f+y0-Y,y0,h);**



> **F(1,0.1);**

**with(VectorCalculus):**

**jac:=Jacobian([f],[Y]);**

**Jac:=unapply(jac,Y);**

**#T:=.001;**

**xold:=1;**

**step:=0.1;**

**for i from 1 to 10 do**

**dx:=LinearAlgebra:-LinearSolve([-Jac(xold)],[F(xold,step)]);**

**#xnew:=xold+dx;**

**#tol:=VectorCalculus[Norm](dx);**

**#xold:=xnew;**

**#if tol < T then break end if**

**end do;**

>

>











Error, (in LinearAlgebra:-LinearSolve) invalid input: LinearAlgebra:-LinearSolve expects its 2nd argument, B, to be of type {Matrix, Vector[column]} but received [-1.1\*Y+1]

> **fsolve(%,Y);**

> **YY[0]:=1.0;**



> **h:=0.1;YY[1]:=fsolve(F(1,0.1),Y=1);**





> **F(YY[1],YY[0],0.1);**



> **for i from 2 to 10 do YY[i]:=fsolve(F(YY[i-1],0.1),Y=YY[i-1]);od;**

>



















> **err:=abs(subs(sol(1),y(t))-YY[10]);**



> **EulerBDF:=proc(f,y00,tf,N)**

**local F,YY,h,y0,i;**

**F:=unapply(h\*f+y0-Y,y0,h);**

**h:=tf/N;**

**YY[0]:=y00;**

**YY[1]:=fsolve(F(YY[0],h),Y=YY[0]);**

**#print(YY[1]);**

**for i from 2 to N do**

**YY[i]:=fsolve(F(YY[i-1],h),Y=YY[i-1]);**

**od;**

**#[seq([i\*h,YY[i]],i=0..N)];#for printing all the values**

**YY[N];**

**end proc;**



> **EulerBDF2:=proc(f,y00,tf,N)**

**local F,YY,h,y0,i;**

**F:=unapply(h\*f+y0-Y,y0,h);**

**h:=tf/N;**

**YY[0]:=y00;**

**YY[1]:=NewtonRhapson(F(YY[0],h),Y=YY[0]);**

**#print(YY[1]);**

**for i from 2 to N do**

**YY[i]:=fsolve(F(YY[i-1],h),Y=YY[i-1]);**

**od;**

**#[seq([i\*h,YY[i]],i=0..N)];#for printing all the values**

**YY[N];**

**end proc;**



> **sol(1);**



> **y00:=1;N:=10;tf:=1.;**







> **EulerBDF(f,y00,tf,N);**



> **for i from 1 to 10 do**

**N:=2^i:**

**YY[i]:=EulerBDF(f,1.0,1.0,N);od;**

>









































> **for i from 1 to 9 do Y1[i]:=abs(YY[i+1]-YY[i]);od;**



















> **for i from 1 to 8 do Y1[i]/Y1[i+1];od;**

















> **restart:**

**N:=10;**

**y00:=[1,1];**

**eq11:=diff(y[1](t),t)=-y[1](t)^2;**

**eq2:=diff(y[2](t),t)=y[1](t)^2-y[2](t);**

**f2:=subs([y[1](t)=Y[1],y[2](t)=Y[2]],[rhs(eq11),rhs(eq2)]);**

**eq3:=[seq(h\*f2[i]+y0[i]-Y[i],i=1..nops(f2))];**

**#F:=unapply(eq3,seq(Y[i],i=1..nops(f2)),seq(y0[i],i=1..nops(f2)),h);**

**F:=unapply(eq3,seq(y0[i],i=1..nops(f2)),h);**

**seq(Y[i],i=1..nops(f2)),seq(y0[i],i=1..nops(f2));**

**F(seq(y00[i],i=1..nops(f2)),0.1);**

**printlevel:=1;**

**for j from 1 to N do**

**YY[0]:=y00;**

**YY[1]:=fsolve(F(YY[0][1],YY[0][2],1/(2^j)),{Y[1]=YY[0][1],Y[2]=YY[0][2]});**

**for i from 2 to 2^j do**

**YY[i]:=fsolve(F(rhs(YY[i-1][1]),rhs(YY[i-1][2]),1/(2^j)),{YY[i-1][1],YY[i-1][2]});end do;**

**j;**

**YYY[j]:=YY[2^j];**

**end do;**

>

>





































































































> **sol2:=dsolve({eq11,eq2,y[1](0)=1,y[2](0)=0},type=numeric):**

> **sol2(1);**



> **printlevel:=1;**

**for i from 1 to N-1 do**

**Z1[i]:={seq(abs(rhs(YYY[i+1][j])-rhs(YYY[i][j])),j=1..nops(f2))};od;**

**Z1[1][1]/Z1[2][1];**























> **for i from 1 to N-2 do**

**ZZ1[i]:={seq(Z1[i][j]/Z1[i+1][j],j=1..nops(f2))};od;**

















> **###################################### P3 ######################################**

> **restart;**

> **Digits:=15;**



> **eq:=diff(y(t),t)=-100\*y(t);**



> **sol:=dsolve({eq,y(0)=1},type=numeric):**

> **sol(1);**



> **sol:=dsolve({eq,y(0)=1},type=numeric,stiff=true):**

**sol(1);**



> **f:=rhs(eq);**



> **f:=subs(y(t)=Y,f);**



> **F:=unapply(f,Y);**



> **YY[0]:=1.0;**



> **h:=0.1;**



> **YY[1]:=YY[0]+h\*F(YY[0]);**



> **eq1:=Y=YY[0]+h\*F(Y);**



> **fsolve(eq1,Y);sol(0.1);**





> **restart;**

> **Y[1]:=Y[0]+h\*lambda\*Y[0];**



> **eq1:=theta=1+z;**



> **z1:=solve(eq1,z);**



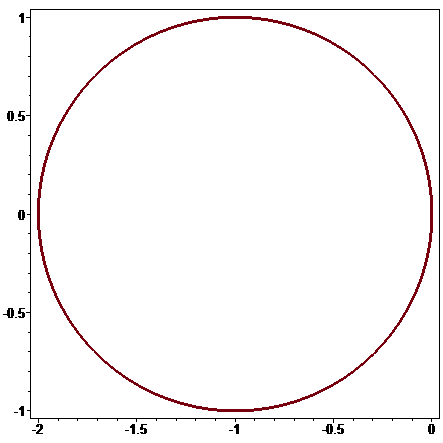
> **z1:=subs(theta=exp(I\*phi),z1);**



> **with(plots):**

**p1:=complexplot(z1,phi=0..2\*Pi,thickness=3,axes=boxed):**

**display({p1});**



> **eq2:=ypred=1+h\*lambda\*1;**



> **eq3:=Y1=1+h/2\*(lambda\*1+lambda\*ypred);**



> **cons:=solve({eq2,eq3},{ypred,Y1});**



> **eqz:=Y1=subs(cons,Y1);**



> **eqz:=subs(Y0=1,eqz);**



> **series(exp(lambda\*h),h);**



> **eqz:=subs(h=z/lambda,Y1=theta,eqz);**



> **solve(eqz,z);**



> **L:=[solve(eqz,z)];**



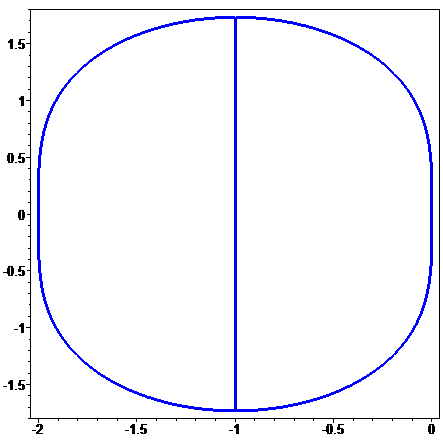
> **L:=subs(theta=exp(I\*phi),L);**



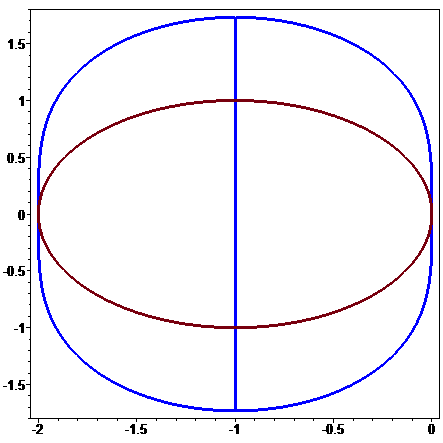
> **p2:=complexplot(L[1],phi=0..2\*Pi,thickness=3,axes=boxed,color=blue):**

**p3:=complexplot(L[2],phi=0..2\*Pi,thickness=3,axes=boxed,color=blue):**

**display({p2,p3});**



> **display({p1,p2,p3});**



> **eq4:=Y1=1+h\*lambda\*Y1;**



> **cons:=solve({eq4},{Y1});**



> **eqz:=Y1=subs(cons,Y1);**



> **eqz:=subs(lambda=z/h,Y1=theta,eqz);**



> **z2:=solve(eqz,z);**

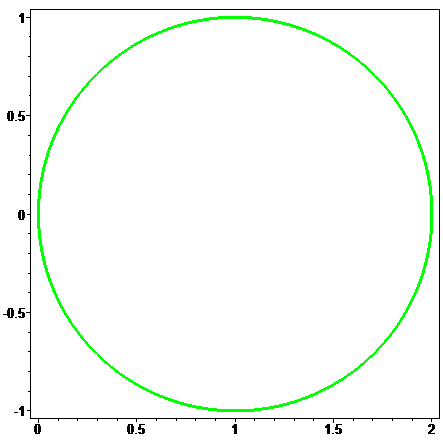


> **z2:=subs(theta=exp(I\*phi),z2);**

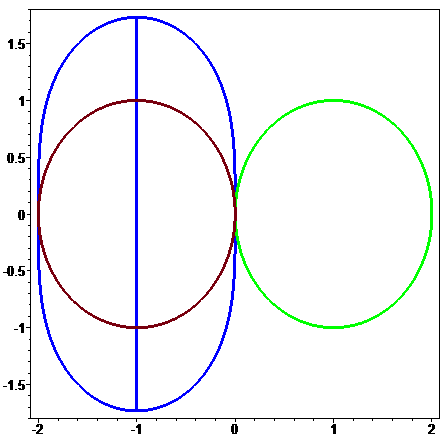


> **p4:=complexplot(z2,phi=0..2\*Pi,thickness=3,axes=boxed,color=green):**

**display({p4});**



> **display({p1,p2,p3,p4});**



> **yy:=y0+b\*t+c\*t^2;**



> **Eq1:=subs(t=-h,yy)=yb;**



> **Eq2:=subs(t=h,diff(yy,t))=F1;**



> **cons:=solve({Eq1,Eq2},{b,c});**



> **subs(cons,t=h,yy);**



> **eq5:=Y1=2/3\*h\*lambda\*Y1+4/3\*1-1/3\*yb;**



> **cons:=solve({eq5},{Y1});**



> **eqz:=Y1=subs(cons,Y1);**



> **eqz:=subs(lambda=z/h,yb=1/theta,Y1=theta,eqz);**



> **z3:=solve(eqz,z);**

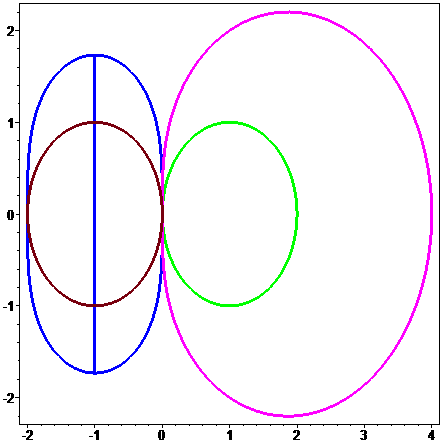


> **z3:=subs(theta=exp(I\*phi),z3);**



> **p5:=complexplot(z3,phi=0..2\*Pi,thickness=3,axes=boxed,color=magenta):**

**display({p1,p2,p3,p4,p5});**



> **yy:=y0+b\*t+c\*t^2;**



> **Eq1:=subs(t=alpha\*h,diff(yy,t))=Fint;**



> **Eq1:=subs(t=h,diff(yy,t))=F1;**



> **cons:=solve({Eq1,Eq2},{b,c});**



> **subs(cons,t=h,yy);**



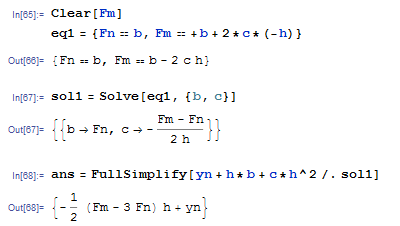
> **eqY:=Y1=subs(cons,t=h,yy);**



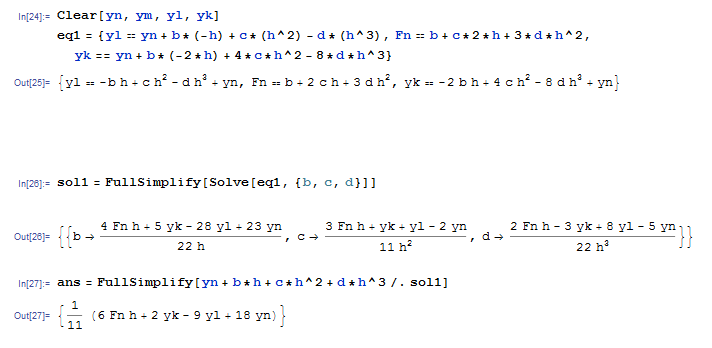
> **eqYint:=Yint=subs(cons,t=alpha\*h,yy);**

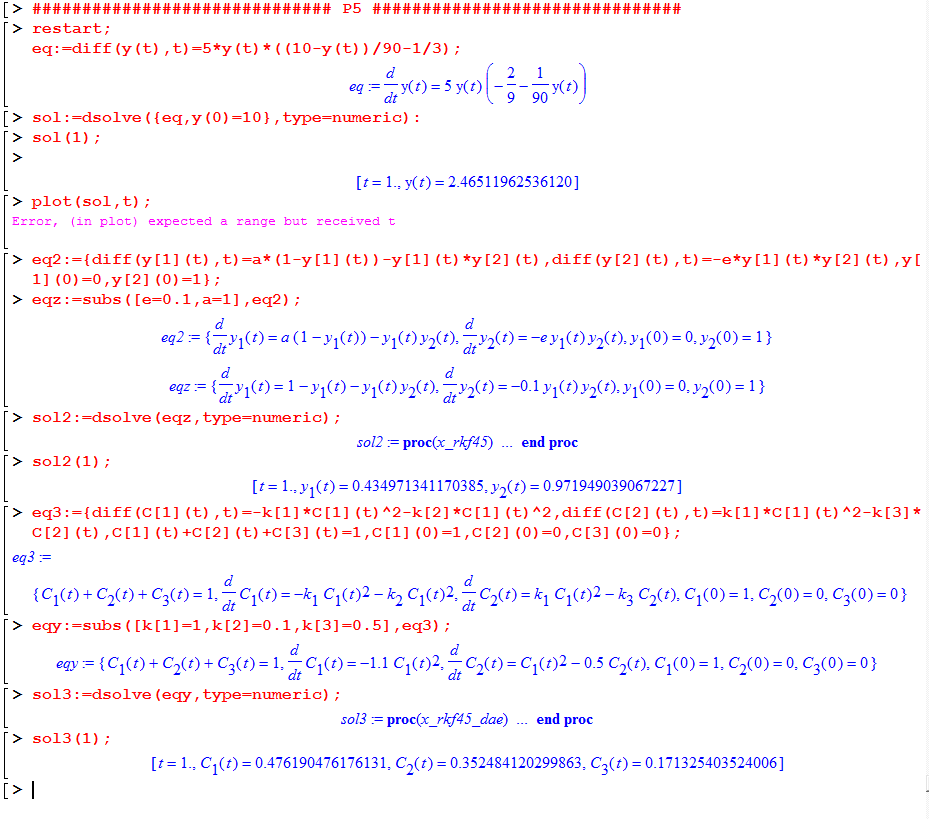


> **############################ P4 #############################**

**Part a)**

**Part b)**





(maple kept crashing on trying to save, so a screenshot is the best I can do for problem 5)