

```
> restart;
```

```
Problem 7.1
```

```
> f := x -> exp(2) * (exp(2*x) - exp(-2*x)) / (exp(4) - 1) + x;
```

$$f := x \rightarrow \frac{e^2 (e^{2x} - e^{-2x})}{e^4 - 1} + x \quad (1)$$

```
> simplify(diff(f(x), x$2) - 4*f(x) + 4*x);
```

0 (2)

```
> f(0); simplify(f(1));
```

0
2 (3)

Note I had the wrong boundary conditions, so wither use the correct ones above or find the exact solution for the boundary conditions $y(0)=1$ and $y(2)=1$.

```
> with(LinearAlgebra):
```

```
> h:=0.05;
```

$h := 0.05$ (4)

```
> xx := Vector(19, i-> i*h);
```

$xx := \begin{bmatrix} 1 \dots 19 \text{ Vector}_{\text{column}} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}$ (5)

```
> y:= Vector(19);
```

```
r:=Vector(19);
```

$y := \begin{bmatrix} 1 \dots 19 \text{ Vector}_{\text{column}} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}$

$r := \begin{bmatrix} 1 \dots 19 \text{ Vector}_{\text{column}} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}$ (6)

```
> #r(1) := -1-4*h^2*xx(1):  
r(19) := -2-4*h^2*xx(19):  
for j from 1 to 18 do  
r(j) := -4*xx(j)*h^2: od:
```

```
> A:= Matrix(19):
```

```
> for j from 1 to 19 do  
A(j,j) := -(2.0+4.0*h^2): od:  
for j from 1 to 18 do  
A(j,j+1) := 1.0:  
A(j+1,j) := 1.0:  
od:
```

```
> y:=LinearSolve(A,r);
```

(7)

$$y := \begin{bmatrix} 1 \dots 19 \text{ Vector}_{\text{column}} \\ \text{Data Type: float}_8 \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix} \quad (7)$$

```
> ff := Vector(19, i-> evalf(f(xx(i))));
```

$$ff := \begin{bmatrix} 1 \dots 19 \text{ Vector}_{\text{column}} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix} \quad (8)$$

```
> evalf(Norm(y-ff,1)/Norm(ff,1));
```

$$0.00008529251978 \quad (9)$$

The other relative error is trivially 0.

Problem 7.2

```
> restart;
```

```
> f := x -> ln(x);
```

$$f := x \rightarrow \ln(x) \quad (10)$$

```
> simplify(diff(f(x), x$2) + diff(f(x), x)^2 + f(x) - ln(x));
```

$$0 \quad (11)$$

```
> f(1); f(2);
```

$$\begin{matrix} 0 \\ \ln(2) \end{matrix} \quad (12)$$

```
> with(LinearAlgebra):
```

```
> h:=0.05;
```

```
yy:= Vector(20):
```

```
ff:= Vector(20):
```

$$h := 0.05 \quad (13)$$

```
> dsys := t-> {diff(y(x), x$2) + diff(y(x), x)^2 + y(x) = ln(x), diff(yt(x), x$2) + 2*diff(yt(x), x)*diff(y(x), x) + yt(x) = 0, y(1) = 0, D(y)(1)=t, yt(1)=0, D(yt)(1)=1};
```

$$dsys := t \rightarrow \left\{ \begin{aligned} \frac{d^2}{dx^2} y(x) + \left(\frac{d}{dx} y(x) \right)^2 + y(x) &= \ln(x), \frac{d^2}{dx^2} yt(x) + 2 \left(\frac{d}{dx} yt(x) \right) \left(\frac{d}{dx} y(x) \right) + yt(x) = 0, \\ y(1) = 0, yt(1) = 0, D(y)(1) = t, D(yt)(1) = 1 \end{aligned} \right\} \quad (14)$$

```
> F:= t->dsolve(dsys(t), numeric, method = rkf45, relerr = 1.*10^(-8), output = listprocedure);
```

$$F := t \rightarrow dsolve \left(dsys(t), \text{numeric}, \text{method} = \text{rkf45}, \text{relerr} = 1. \frac{1}{100000000}, \text{output} = \text{listprocedure} \right) \quad (15)$$

```
> F(0.3);
```

$$\left[x = \text{proc}(x) \dots \text{end proc}, y(x) = \text{proc}(x) \dots \text{end proc}, \frac{d}{dx} y(x) = \text{proc}(x) \dots \text{end proc}, yt(x) = \text{proc}(x) \dots \text{end proc} \right] \quad (16)$$

...

$$\text{end proc}, \frac{d}{dx} yt(x) = \text{proc}(x) \dots \text{end proc}]$$

```
> g := t-> eval(y(x), F(t))-ln(2.0);
gt := t -> eval(yt(x), F(t));
g := t->eval(y(x), F(t)) - ln(2.0)
gt := t->eval(yt(x), F(t))
```

(17)

```
> g(0.3) (2.0);gt(0.3) (2.0);
-0.357505648083878
0.627635180894839
```

(18)

```
> t:=0.3;
for j from 1 to 8 do
t := t- g(t) (2.0)/gt(t) (2.0);
od;
g(t) (2.0);

t := 0.3
t := 0.869607407242805
t := 0.995889156881516
t := 0.999995999176674
t := 1.00000000671389
t := 1.00000000671573
t := 1.00000000671576
t := 1.00000000671580
t := 1.00000000671575
-7.54951656745106 10-15
```

(19)

relative error at x=2.0

```
> abs(g(t) (2.0))/ln(2.0);
1.08916501123413 10-14
```

(20)

```
> for j from 1 to 20 do
z:= eval(y(x), F(t)) (1.0+j*h);
yy(j) :=z;
ff(j) := ln(1+j*h);
od;
> Norm(yy-ff,1)/Norm(ff,1);
3.080583854 10-9
```

(21)

```
> yy-ff;
[ 1 .. 20 Vectorcolumn
Data Type: float8
Storage: rectangular
Order: Fortran_order ]
```

(22)

```
> abs(yy(20)-ln(2.0))/ln(2.0);
1.08916501123413 10-14
```

(23)