```
> restart;
```

Problem 10.1

```
> f:= x-> exp(1+x^2);
```

$$f := x \to e^{1+x^2} \tag{1}$$

```
> -diff(f(x),x$2)+2*f(x)+4*x^2*exp(1+x^2);
```

$$0 \tag{2}$$

```
> simplify(%);
```

$$0 \tag{3}$$

```
> a:= 0;
  b:=1;
  alpha:= evalf(exp(1.0));
  beta:= evalf(exp(2.0));
  p := x-> 1;
  q:= x-> 2;
  g:= x->-4*x^2*exp(1+x^2);
  L:= u->-diff((p(x)*diff(u(x),x)),x)+q(x)*u(x);
  U:= f;#solution
  simplify(L(U)-g(x));
```

$$a := 0$$

$$b := 1$$

$$\alpha := 2.718281828$$

$$\beta := 7.389056099$$

$$p := x \to 1$$

$$q := x \to 2$$

$$g := x \to -4\,x^2\,e^{1+x^2}$$

$$L := u \to -\left(\frac{d}{dx}\left(p(x)\left(\frac{d}{dx}\,u(x)\right)\right)\right) + q(x)\,u(x)$$

$$U := f$$

$$0 \tag{4}$$

```
> l:= x-> alpha+(x-a)/(b-a)*(beta-alpha);
```

$$l := x \to \alpha + \frac{(x-a)\,(\beta-\alpha)}{b-a} \tag{5}$$

```
> L(U-l);
  evalf(U(a))-l(a);
  evalf(U(b))-l(b);
```

$$-4\,x^2\,e^{1+x^2} - 5.436563656 - 9.341548542\,x$$

$$0.$$

$$0. \tag{6}$$

```
> lprint(L(U-l));
-4*x^2*exp(1+x^2)-5.436563656-9.341548542*x
```

New RHS of diff eq with 0 boundary values.

```
> G:= x-> -4*x^2*exp(1+x^2)-5.436563656-9.341548542*x;#new RHS of diff eq
```

$$G := x \to -4\,x^2\,e^{1+x^2} - 5.436563656 - 9.341548542\,x \tag{7}$$

```
> simplify(L(U-l)-G(x));
```

$$0. \tag{8}$$

```
> with(CurveFitting);
```

$$[ArrayInterpolation, BSpline, BSplineCurve, Interactive, LeastSquares, PolynomialInterpolation, RationalInterpolation, Spline, ThieleInterpolation] \tag{9}$$

```
> with(LinearAlgebra):
> N:= 10;
```

```
  h:=1.0/(N+1);
> for j from 1 to N do
  kn||j:= [a+(j-1)*h,a+j*h,a+(j+1)*h];
  od;
> for j from 1 to N do
  phi||j:= unapply(BSpline(2,x,knots=kn||j),x):
  od:
  M:= Matrix(N):
  for j from 1 to N do for i from 1 to N do
  if (abs(j-i)<=1.1) then M(i,j):= int(p(x)*diff(phi||j(x),x)*diff(phi||i(x),
  x),x=a+min(i-1,j-1)*h..a+max(i+1,j+1)*h)
  +int(q(x)*phi||j(x)*phi||i(x),x=a+min(i-1,j-1)*h..a+max(i+1,j+1)*h): fi:
  od:
  od:
```

$$N := 10$$

$$h := 0.09090909091$$

$$kn1 := [0., 0.09090909091, 0.1818181818]$$

$$kn2 := [0.09090909091, 0.1818181818, 0.2727272727]$$

$$kn3 := [0.1818181818, 0.2727272727, 0.3636363636]$$

$$kn4 := [0.2727272727, 0.3636363636, 0.4545454546]$$

$$kn5 := [0.3636363636, 0.4545454546, 0.5454545455]$$

$$kn6 := [0.4545454546, 0.5454545455, 0.6363636364]$$

$$kn7 := [0.5454545455, 0.6363636364, 0.7272727273]$$

$$kn8 := [0.6363636364, 0.7272727273, 0.8181818182]$$

$$kn9 := [0.7272727273, 0.8181818182, 0.9090909091]$$

$$kn10 := [0.8181818182, 0.9090909091, 1.000000000] \quad (10)$$

```
> M;
```

$$[[22.12121212, -10.96969697, 0, 0, 0, 0, 0, 0, 0, 0], \quad (11)$$
$$[-10.96969697, 22.12121212, -10.96969697, 0, 0, 0, 0, 0, 0, 0],$$
$$[0, -10.96969697, 22.12121212, -10.96969697, 0, 0, 0, 0, 0, 0],$$
$$[0, 0, -10.96969697, 22.12121212, -10.96969697, 0, 0, 0, 0, 0],$$
$$[0, 0, 0, -10.96969697, 22.12121213, -10.96969698, 0, 0, 0, 0],$$
$$[0, 0, 0, 0, -10.96969698, 22.12121212, -10.96969697, 0, 0, 0],$$
$$[0, 0, 0, 0, 0, -10.96969697, 22.12121212, -10.96969697, 0, 0],$$
$$[0, 0, 0, 0, 0, 0, -10.96969697, 22.12121212, -10.96969697, 0],$$
$$[0, 0, 0, 0, 0, 0, 0, -10.96969697, 22.12121212, -10.96969697],$$
$$[0, 0, 0, 0, 0, 0, 0, 0, -10.96969697, 22.12121212]]$$

```
> v:=Vector(N):
  for i from 1 to N do
  v(i):= int(G(x)*phi||i(x),x=a+(i-1)*h..a+(i+1)*h):
  od:
> v;
```

(12)

$$\begin{bmatrix} -0.5811074074 \\ -0.6840623155 \\ -0.8070767595 \\ -0.9548749024 \\ -1.134907849 \\ -1.358351640 \\ -1.641655044 \\ -2.008905783 \\ -2.495441393 \\ -3.153381281 \end{bmatrix} \quad (12)$$

```
> trueSol:=Vector(N):
  for j from 1 to N do
  trueSol(j):= evalf(f(a+j*h)):
  od:
```

```
> solution:= LinearSolve(M,v);
  for j from 1 to N do
  solution(j):= solution(j)+evalf(l(a+j*h)):
  od:
```

$$solution := \begin{bmatrix} -0.402512040114266 \\ -0.758721670761122 \\ -1.06514752673406 \\ -1.31565446567262 \\ -1.50092122487789 \\ -1.60760668078976 \\ -1.61710991397774 \\ -1.50376243704436 \\ -1.23220690401878 \\ -0.753589700781183 \end{bmatrix} \quad (13)$$

```
> Norm(solution-trueSol)/Norm(trueSol);
  Norm(solution-trueSol,1)/Norm(trueSol,1);
```

$$0.0003019302429$$
$$0.0003498543714 \quad (14)$$

```
> N:= 20;
  h:=1.0/(N+1);
```

$$N := 20$$
$$h := 0.04761904762 \quad (15)$$

```
> for j from 1 to N do
  kn||j:= [a+(j-1)*h,a+j*h,a+(j+1)*h];
  od;
> for j from 1 to N do
  phi||j:= unapply(BSpline(2,x,knots=kn||j),x):
  od:
  M:= Matrix(N):
  for j from 1 to N do for i from 1 to N do
  if (abs(j-i)<=1.1) then M(i,j):= int(p(x)*diff(phi||j(x),x)*diff(phi||i(x),
```

```
x),x=a+min(i-1,j-1)*h..a+max(i+1,j+1)*h)
+int(q(x)*phi||j(x)*phi||i(x),x=a+min(i-1,j-1)*h..a+max(i+1,j+1)*h): fi:
od:
od:
```

$$kn1 := [0., 0.04761904762, 0.09523809524]$$
$$kn2 := [0.04761904762, 0.09523809524, 0.1428571429]$$
$$kn3 := [0.09523809524, 0.1428571429, 0.1904761905]$$
$$kn4 := [0.1428571429, 0.1904761905, 0.2380952381]$$
$$kn5 := [0.1904761905, 0.2380952381, 0.2857142857]$$
$$kn6 := [0.2380952381, 0.2857142857, 0.3333333333]$$
$$kn7 := [0.2857142857, 0.3333333333, 0.3809523810]$$
$$kn8 := [0.3333333333, 0.3809523810, 0.4285714286]$$
$$kn9 := [0.3809523810, 0.4285714286, 0.4761904762]$$
$$kn10 := [0.4285714286, 0.4761904762, 0.5238095238]$$
$$kn11 := [0.4761904762, 0.5238095238, 0.5714285714]$$
$$kn12 := [0.5238095238, 0.5714285714, 0.6190476191]$$
$$kn13 := [0.5714285714, 0.6190476191, 0.6666666667]$$
$$kn14 := [0.6190476191, 0.6666666667, 0.7142857143]$$
$$kn15 := [0.6666666667, 0.7142857143, 0.7619047619]$$
$$kn16 := [0.7142857143, 0.7619047619, 0.8095238095]$$
$$kn17 := [0.7619047619, 0.8095238095, 0.8571428572]$$
$$kn18 := [0.8095238095, 0.8571428572, 0.9047619048]$$
$$kn19 := [0.8571428572, 0.9047619048, 0.9523809524]$$
$$kn20 := [0.9047619048, 0.9523809524, 1.000000000] \quad (16)$$

```
> M;
```

$$\begin{bmatrix} 20 \text{ x } 20 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix} \quad (17)$$

```
> v:=Vector(N):
  for i from 1 to N do
  v(i):= int(G(x)*phi||i(x),x=a+(i-1)*h..a+(i+1)*h):
  od:
> v;
```

$$\begin{bmatrix} 1 \text{ .. } 20 \text{ Vector}_{column} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix} \quad (18)$$

```
> trueSol:=Vector(N):
  for j from 1 to N do
  trueSol(j):= evalf(f(a+j*h)):
  od:

> solution:= LinearSolve(M,v);
  for j from 1 to N do
  solution(j):= solution(j)+evalf(l(a+j*h)):
  od:
```

$$solution := \begin{bmatrix} 1 \,..\, 20\ Vector_{column} \\ Data\ Type: float_8 \\ Storage: rectangular \\ Order: Fortran\_order \end{bmatrix} \tag{19}$$

```
> Norm(solution-trueSol)/Norm(trueSol);
  Norm(solution-trueSol,1)/Norm(trueSol,1);
```

$$0.00007626052033$$
$$0.00009088982029 \tag{20}$$

Problem 10.2

```
> restart;
> with(LinearAlgebra);
  v := Vector(2);
  w := Vector(2);
```

[&x, Add, Adjoint, BackwardSubstitute, BandMatrix, Basis, BezoutMatrix, BidiagonalForm, BilinearForm, CARE, CharacteristicMatrix, CharacteristicPolynomial, Column, ColumnDimension, ColumnOperation, ColumnSpace, CompanionMatrix, ConditionNumber, ConstantMatrix, ConstantVector, Copy, CreatePermutation, CrossProduct, DARE, DeleteColumn, DeleteRow, Determinant, Diagonal, DiagonalMatrix, Dimension, Dimensions, DotProduct, EigenConditionNumbers, Eigenvalues, Eigenvectors, Equal, ForwardSubstitute, FrobeniusForm, GaussianElimination, GenerateEquations, GenerateMatrix, Generic, GetResultDataType, GetResultShape, GivensRotationMatrix, GramSchmidt, HankelMatrix, HermiteForm, HermitianTranspose, HessenbergForm, HilbertMatrix, HouseholderMatrix, IdentityMatrix, IntersectionBasis, IsDefinite, IsOrthogonal, IsSimilar, IsUnitary, JordanBlockMatrix, JordanForm, KroneckerProduct, LA_Main, LUDecomposition, LeastSquares, LinearSolve, LyapunovSolve, Map, Map2, MatrixAdd, MatrixExponential, MatrixFunction, MatrixInverse, MatrixMatrixMultiply, MatrixNorm, MatrixPower, MatrixScalarMultiply, MatrixVectorMultiply, MinimalPolynomial, Minor, Modular, Multiply, NoUserValue, Norm, Normalize, NullSpace, OuterProductMatrix, Permanent, Pivot, PopovForm, QRDecomposition, RandomMatrix, RandomVector, Rank, RationalCanonicalForm, ReducedRowEchelonForm, Row, RowDimension, RowOperation, RowSpace, ScalarMatrix, ScalarMultiply, ScalarVector, SchurForm, SingularValues, SmithForm, StronglyConnectedBlocks, SubMatrix, SubVector, SumBasis, SylvesterMatrix, SylvesterSolve, ToeplitzMatrix, Trace, Transpose, TridiagonalForm, UnitVector, VandermondeMatrix, VectorAdd, VectorAngle, VectorMatrixMultiply, VectorNorm, VectorScalarMultiply, ZeroMatrix, ZeroVector, Zip]

$$v := \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$w := \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{21}$$

```
> f1:=x1^2/9+x2^2/16-1:
  f2:=x1^2/16+x2^2/9-1:
> F := proc(f,N,v)
  with(LinearAlgebra);
  w:= Vector(N);
  for k from 1 to N do
  w(k):=f||k;
  for j from 1 to N do
  w(k):= subs(x||j=v(j),w(k)):
  od:
  od:
```

```
  w;
  end proc;
```
Warning, `w` is implicitly declared local to procedure `F`
Warning, `k` is implicitly declared local to procedure `F`
Warning, `j` is implicitly declared local to procedure `F`

$$F := \mathbf{proc}(f, N, v) \quad (22)$$

```
    local w, k, j;
    with(LinearAlgebra);
    w := Vector(N);
    for k to N do w(k) := f||k; for j to N do w(k) := subs(x||j = v(j), w(k)) end do end do;
    w
end proc
```

```
> F(f,2,v);
```

$$\begin{bmatrix} -1 \\ -1 \end{bmatrix} \quad (23)$$

```
> JF := proc(f,N,v)
  with(LinearAlgebra);
  Jac:= Matrix(N);
  for a from 1 to N do
  for b from 1 to N do
  Jac(a,b):= diff(f||a,x||b):
  od:
  od:
  #print(Jac);
  for a from 1 to N do
  for b from 1 to N do
  for j from 1 to N do
  Jac(a,b):= subs(x||j=v(j),Jac(a,b)):
  od:
  od:
  od:
  Jac;
  end proc;
```
Warning, `Jac` is implicitly declared local to procedure `JF`
Warning, `a` is implicitly declared local to procedure `JF`
Warning, `b` is implicitly declared local to procedure `JF`
Warning, `j` is implicitly declared local to procedure `JF`

$$JF := \mathbf{proc}(f, N, v) \quad (24)$$

```
    local Jac, a, b, j;
    with(LinearAlgebra);
    Jac := Matrix(N);
    for a to N do for b to N do Jac(a, b) := diff(f||a, x||b) end do end do;
    for a to N do
        for b to N do for j to N do Jac(a, b) := subs(x||j = v(j), Jac(a, b)) end do end do
    end do;
    Jac
end proc
```

```
> v(1):=1;
  v(2):=1;
```

$$v := \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$v := \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{25}$$

```
> Newton2:= proc (v,f,N)
  with(LinearAlgebra):
  w:= Vector(N);
  w:= v-MatrixInverse(JF(f,2,v)) . F(f,2,v);
  w;
  end proc;
```

Warning, `w` is implicitly declared local to procedure `Newton2`

$$\begin{aligned} &Newton2 := \mathbf{proc}(v, f, N) \\ &\quad \mathbf{local}\ w; \\ &\quad with(LinearAlgebra); \\ &\quad w := Vector(N); \\ &\quad w := v - \text{`.`}(LinearAlgebra:\text{-}MatrixInverse(JF(f, 2, v)), F(f, 2, v)); \\ &\quad w \\ &\mathbf{end\ proc} \end{aligned} \tag{26}$$

```
> for j from 1 to 5 do
  v:= evalf(Newton2(v,f,2));
  od;
```

$$v := \begin{bmatrix} 3.380000000 \\ 3.380000000 \end{bmatrix}$$

$$v := \begin{bmatrix} 2.54207100600000 \\ 2.54207100600000 \end{bmatrix}$$

$$v := \begin{bmatrix} 2.40397002500000 \\ 2.40397002500000 \end{bmatrix}$$

$$v := \begin{bmatrix} 2.40000327800000 \\ 2.40000327800000 \end{bmatrix}$$

$$v := \begin{bmatrix} 2.40000000000000 \\ 2.40000000000000 \end{bmatrix} \tag{27}$$

```
> v(1)^2/16+v(2)^2/9-1;
```

$$-1.11022302462516\ 10^{-16} \tag{28}$$