```
> restart;
```

Homework 5

The following procedure compute R_{nn} for the integral of f(x) over [a,b]

```
> with(Student[Calculus1]);

  Romberg := proc(f,a,b,n)
  local k,j,R,Romb; # n > 0; needs Student[Calculus1]
  R[1][1] := ApproximateInt(f(x), x = a..b,method=trapezoid,
  partition=1):
  for j from 2 to n do
  R[j][1] := ApproximateInt(f(x), x = a..b,method=trapezoid,
  partition=2^(j-1)):
  for k from 2 to j do
  R[j][k]:= (2^k*R[j][k-1]-R[j-1][k-1])/(2^k-1);
  od;
  od;
  for j from 1 to n do
  Romb[j]:=R[j][j];
  od;
  return Romb
  end proc;
```

[*AntiderivativePlot*, *AntiderivativeTutor*, *ApproximateInt*, *ApproximateIntTutor*, *ArcLength*, *ArcLengthTutor*, *Asymptotes*, *Clear*, *CriticalPoints*, *CurveAnalysisTutor*, *DerivativePlot*, *DerivativeTutor*, *DiffTutor*, *ExtremePoints*, *FunctionAverage*, *FunctionAverageTutor*, *FunctionChart*, *FunctionPlot*, *GetMessage*, *GetNumProblems*, *GetProblem*, *Hint*, *InflectionPoints*, *IntTutor*, *Integrand*, *InversePlot*, *InverseTutor*, *LimitTutor*, *MeanValueTheorem*, *MeanValueTheoremTutor*, *NewtonQuotient*, *NewtonsMethod*, *NewtonsMethodTutor*, *PointInterpolation*, *RiemannSum*, *RollesTheorem*, *Roots*, *Rule*, *Show*, *ShowIncomplete*, *ShowSolution*, *ShowSteps*, *Summand*, *SurfaceOfRevolution*, *SurfaceOfRevolutionTutor*, *Tangent*, *TangentSecantTutor*, *TangentTutor*, *TaylorApproximation*, *TaylorApproximationTutor*, *Understand*, *Undo*, *VolumeOfRevolution*, *VolumeOfRevolutionTutor*, *WhatProblem*]

```
Romberg := proc(f, a, b, n)                                                    (1)
    local k, j, R, Romb;
    R[1][1] := Student:-Calculus1:-ApproximateInt(f(x), x = a..b, method = trapezoid,
    partition = 1);
    for j from 2 to n do
        R[j][1] := Student:-Calculus1:-ApproximateInt(f(x), x = a..b, method = trapezoid,
        partition = 2^(j - 1));
        for k from 2 to j do
            R[j][k] := (2^k * R[j][k - 1] - R[j - 1][k - 1]) / (2^k - 1)
        end do
    end do;
    for j to n do Romb[j] := R[j][j] end do;
    return Romb
end proc
```

pg 217 Problems 1a, 3a, and 5a

```
> f := x -> x^2*ln(x);
```

$$f := x \to x^2 \ln(x) \quad \textbf{(2)}$$

```
> R:=Romberg(f,1,1.5,10);
```

$$R := Romb \quad \textbf{(3)}$$

```
> R[3];
  R[4];
```

0.1922603396

0.1922593601 **(4)**

Note n = 4 suffices in problem 5a

```
> Digits:=15;
  R:=Romberg(f,1,1.5,10);
  for j from 2 to 10
  do
  print(j,R[j],abs(R[j-1]-R[j]), abs(R[j]-int(f(x),x=1..1.5)));
  od;
```

$Digits := 15$

$R := Romb$

2, 0.192245307413098, 0.035828815897744, 0.000014050319698

3, 0.192260339450254, 0.000015032037156, $9.81717458\ 10^{-7}$

4, 0.192259360255601, $9.79194653\ 10^{-7}$, $2.522805\ 10^{-9}$

5, 0.192259357695770, $2.559831\ 10^{-9}$, $3.7026\ 10^{-11}$

6, 0.192259357732695, $3.6925\ 10^{-11}$, $1.01\ 10^{-13}$

7, 0.192259357732798, $1.03\ 10^{-13}$, $2.\ 10^{-15}$

8, 0.192259357732795, $3.\ 10^{-15}$, $1.\ 10^{-15}$

9, 0.192259357732797, $2.\ 10^{-15}$, $1.\ 10^{-15}$

10, 0.192259357732795, $2.\ 10^{-15}$, $1.\ 10^{-15}$ **(5)**

pg 217 Problems 1b, 3b, and 5b

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

$$f := x \to x^2 e^{-x} \quad \textbf{(6)}$$

```
> Digits:=10;
  R:=Romberg(f,0.0,1.0,10);
```

$Digits := 10$

$R := Romb$ **(7)**

```
> R[3];
  R[4];
```

0.1604825891

0.1606018341 **(8)**

Note n = 5 suffices in problem 5b

```
> Digits:=15;
  R:=Romberg(f,0.0,1.0,10);
  for j from 2 to 10
  do
  print(j,R[j],abs(R[j-1]-R[j]), abs(R[j]-int(f(x),x=1..1.5)));
  od;
```

$Digits := 15$

$R := Romb$

2, 0.162401683480680, 0.021538037105041, 0.059301861300415

3, 0.160482589070654, 0.001919094410026, 0.061220955710441

4, 0.160601834058778, 0.000119244988124, 0.061101710722317

5, 0.160602808787224, $9.74728446\ 10^{-7}$, 0.061100735993871

6, 0.160602794166002, $1.4621222\ 10^{-8}$, 0.061100750615093

7, 0.160602794142699, $2.3303\ 10^{-11}$, 0.061100750638396

8, 0.160602794142788, $8.9\ 10^{-14}$, 0.061100750638307

9, 0.160602794142789, $1.\ 10^{-15}$, 0.061100750638306

10, 0.160602794142787, $2.\ 10^{-15}$, 0.061100750638308 **(9)**

pg 226 Problem 1a

```
> with(Student[Calculus1]);
```

[*AntiderivativePlot, AntiderivativeTutor, ApproximateInt, ApproximateIntTutor, ArcLength, ArcLengthTutor, Asymptotes, Clear, CriticalPoints, CurveAnalysisTutor, DerivativePlot, DerivativeTutor, DiffTutor, ExtremePoints, FunctionAverage, FunctionAverageTutor, FunctionChart, FunctionPlot, GetMessage, GetNumProblems, GetProblem, Hint, InflectionPoints, IntTutor, Integrand, InversePlot, InverseTutor, LimitTutor, MeanValueTheorem, MeanValueTheoremTutor, NewtonQuotient, NewtonsMethod, NewtonsMethodTutor, PointInterpolation, RiemannSum, RollesTheorem, Roots, Rule, Show, ShowIncomplete, ShowSolution, ShowSteps, Summand, SurfaceOfRevolution, SurfaceOfRevolutionTutor, Tangent, TangentSecantTutor, TangentTutor, TaylorApproximation, TaylorApproximationTutor, Understand, Undo, VolumeOfRevolution, VolumeOfRevolutionTutor, WhatProblem*] **(10)**

```
> f:= x -> x^2*ln(x);
```

$f := x \rightarrow x^2 \ln(x)$ **(11)**

```
> a:=1.0;
  b:=1.5;
  TrueValue:= int(f(x),x=a..b);
```

$a := 1.0$

$b := 1.5$

$TrueValue := 0.192259357732796$ **(12)**

```
> Sab := ApproximateInt(f(x),x=a..b,method=simpson, partition=1);
  S1  := ApproximateInt(f(x),x=a..(a+b)/2,method=simpson,
  partition=1);
  S2   := ApproximateInt(f(x),x=(a+b)/2..b,method=simpson,
  partition=1);
```

$Sab := 0.192245307413098$

$S1 := 0.0393724340391206$

$S2 := 0.152886026406489$ **(13)**

```
> ErrEst:=abs(Sab-S1-S2)/15.0;
  abs(S1+S2-TrueValue);
```

$ErrEst := 8.76868834133334\ 10^{-7}$

$8.97287186\ 10^{-7}$ **(14)**

pg 226 Problem 2a

```
> f:= x -> exp(3*x)*sin(2*x);
```

$$f := x \to e^{3x} \sin(2x) \quad (15)$$

```
> a:=0.0;
  b:=Pi/4.0;
  TrueValue:= int(f(x),x=a..b);
```

$$a := 0.$$

$$b := 0.785398163397448$$

$$TrueValue := 2.58862863250717 \quad (16)$$

```
> Sab := ApproximateInt(f(x),x=a..b,method=simpson, partition=1);
  S1  := ApproximateInt(f(x),x=a..(a+b)/2,method=simpson,
  partition=1);
  S2   := ApproximateInt(f(x),x=(a+b)/2..b,method=simpson,
  partition=1);
```

$$Sab := 2.58369640324748$$

$$S1 := 0.330889269595191$$

$$S2 := 2.25681218386343 \quad (17)$$

```
> ErrEst:=abs(Sab-S1-S2)/15.0;
  abs(S1+S2-TrueValue);
```

$$ErrEst := 0.000267003347409333$$

$$0.00092717904855 \quad (18)$$

Note that in 1a and 2a, the estimate is almost as good as the true error (even though we are using a rather large interval)

Problem 3a Tol = 10^(-3)

```
> f:= x -> x^2*ln(x);
```

$$f := x \to x^2 \ln(x) \quad (19)$$

```
> a:=1.0;
  b:=1.5;
  TrueValue:= int(f(x),x=a..b);
  Tol := 10^(-3);
```

$$a := 1.0$$

$$b := 1.5$$

$$TrueValue := 0.192259357732796$$

$$Tol := \frac{1}{1000} \quad (20)$$

```
> Sab := ApproximateInt(f(x),x=a..b,method=simpson, partition=1);
  S1  := ApproximateInt(f(x),x=a..(a+b)/2,method=simpson,
  partition=1);
  S2   := ApproximateInt(f(x),x=(a+b)/2..b,method=simpson,
  partition=1);
```

$$Sab := 0.192245307413098$$

$$S1 := 0.0393724340391206$$

$$S2 := 0.152886026406489 \quad (21)$$

```
> ErrEst:=abs(Sab-S1-S2)/15.0;
```

$$ErrEst := 8.76868834133334\ 10^{-7} \quad (22)$$

This is < Tol, so we are done.

Problem 6a with Tol 10^(-4) (instead of 10^(-5))

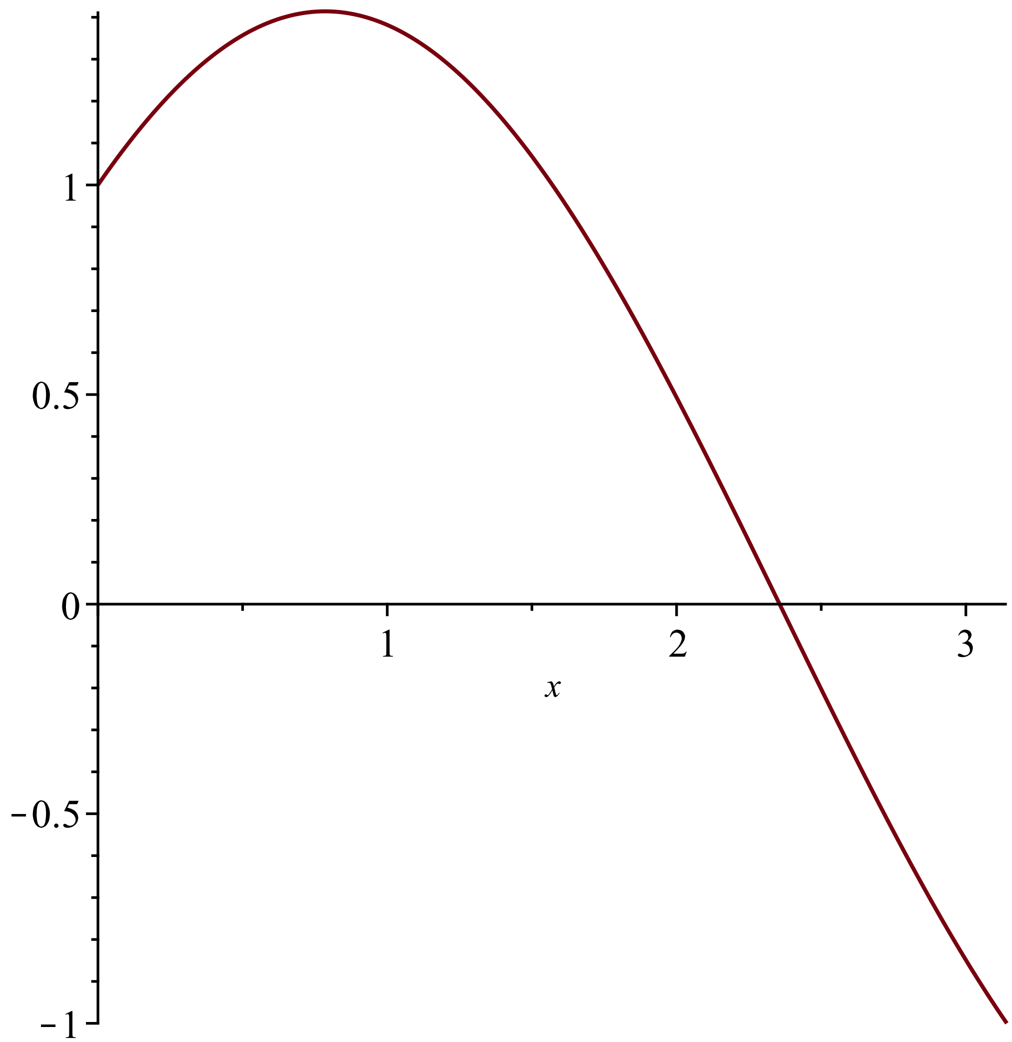
```
> f:= x -> sin(x)+cos(x);
```

$$f := x \rightarrow \sin(x) + \cos(x) \tag{23}$$

```
> a:=0.0;
  b:=evalf(Pi);
  plot(f(x),x=a..b);
  TrueValue:= int(f(x),x=a..b);
  Tol := 10^(-4);
```

$$a := 0.$$

$$b := 3.14159265358979$$



$$TrueValue := 2.00000000000000$$

$$Tol := \frac{1}{10000} \tag{24}$$

```
> S1 := ApproximateInt(f(x),x=a..b,method=simpson, partition=1);
  midpoint1:=(a+b)/2:
  S11  := ApproximateInt(f(x),x=a..midpoint1,method=simpson,
  partition=1);
  S12   := ApproximateInt(f(x),x=midpoint1..b,method=simpson,
  partition=1);
```

$$S1 := 2.09439510239320$$

$$S11 := 2.00455975498442$$

$$S12 := -1.41371669411540\ 10^{-16} \tag{25}$$

```
> ErrEst1 :=abs(S1-S11-S12)/15.0;
```

$$ErrEst1 := 0.00598902316058534 \tag{26}$$

Not within Tol, so subdivide each half into two pieces and use Tol/2

```
> Tol:=Tol/2.0;
  S11   := ApproximateInt(f(x),x=a..midpoint1,method=simpson,
  partition=1);
  midpoint11:= (a+midpoint1)/2:
  S111  := ApproximateInt(f(x),x=a..midpoint11,method=simpson,
  partition=1);
  S112    := ApproximateInt(f(x),x=midpoint11..midpoint1,method=
  simpson, partition=1);
  midpoint12:= (midpoint1+b)/2:
  S12  := ApproximateInt(f(x),x=midpoint1..b,method=simpson,
  partition=1);
  S121   := ApproximateInt(f(x),x=midpoint1..midpoint12,method=
  simpson, partition=1);
  S122    := ApproximateInt(f(x),x=midpoint12..b,method=simpson,
  partition=1);
```

$$Tol := 0.0000500000000000000$$

$$S11 := 2.00455975498442$$

$$S111 := 1.00013458497420$$

$$S112 := 1.00013458497419$$

$$S12 := -1.41371669411540\ 10^{-16}$$

$$S121 := 0.414269309294694$$

$$S122 := -0.414269309294695 \tag{27}$$

```
> ErrEst11:=abs(S11-S111-S112)/15.0;
  ErrEst12:=abs(S12-S121-S122)/15.0;
```

$$ErrEst11 := 0.000286039002402000$$

$$ErrEst12 := 6.66666666666667\ 10^{-17} \tag{28}$$

S121+S122 is within Tol/2, but S121+S122 is not, so subdivide the right half interval, and halve Tol

```
> Tol:=Tol/2.0;
  S111   := ApproximateInt(f(x),x=a..midpoint11,method=simpson,
  partition=1);
  midpoint111:= (a+midpoint11)/2:
  S1111  := ApproximateInt(f(x),x=a..midpoint111,method=simpson,
  partition=1);
  S1112    := ApproximateInt(f(x),x=midpoint111..midpoint11,method=
  simpson, partition=1);
  midpoint112:= (midpoint11+midpoint1)/2:
  S112  := ApproximateInt(f(x),x=midpoint11..midpoint1,method=
  simpson, partition=1);
  S1121   := ApproximateInt(f(x),x=midpoint11..midpoint112,method=
  simpson, partition=1);
  S1122    := ApproximateInt(f(x),x=midpoint112..midpoint1,method=
  simpson, partition=1);
```

$$Tol := 0.0000250000000000000$$

$$S111 := 1.00013458497420$$

$$S1111 := 0.458807705872552$$

$$S1112 := 0.541200589651418$$
$$S112 := 1.00013458497419$$
$$S1121 := 0.541200589651424$$
$$S1122 := 0.458807705872544 \quad \textbf{(29)}$$

```
> ErrEst111:=abs(S111-S1111-S1112)/15.0;
  ErrEst112:=abs(S112-S1121-S1122)/15.0;
```

$$ErrEst111 := 0.00000841929668200000$$
$$ErrEst112 := 0.00000841929668146667 \quad \textbf{(30)}$$

Both are with our Tol = (10^(-4))/4, so we ca settle for the approximation

```
> OurApprox := S121+S122+ S1111+S1112 +S1121+S1122;
```

$$OurApprox := 2.00001659104793 \quad \textbf{(31)}$$

```
> abs(TrueValue-OurApprox);
```

$$0.00001659104793 \quad \textbf{(32)}$$

pg 234 Problems 1a and 1c
scaling from [-1,1] to [a,b] is done by x=(b-a)/2*t+(b+a)/2.
Weights get multiplied by (b-a).

```
> f:= x -> x^2*ln(x);
  a:=1.0;
  b:=1.5;
  u:=t -> (b-a)/2*t+(a+b)/2;
  x1:=u(-1/sqrt(3.0)); x2:=u(1/sqrt(3.0));
  w1:=(b-a)/2; w2:=(b-a)/2;
  TrueValue:= int(f(x),x=a..b);
  Approx:= w1*f(x1) + w2*f(x2);
  AbsoluteError:=abs(TrueValue-Approx);
```

$$f := x \to x^2 \ln(x)$$
$$a := 1.0$$
$$b := 1.5$$
$$u := t \to \frac{1}{2}(b-a)\,t + \frac{1}{2}a + \frac{1}{2}b$$
$$x1 := 1.10566243270259$$
$$x2 := 1.39433756729741$$
$$w1 := 0.250000000000000$$
$$w2 := 0.250000000000000$$
$$TrueValue := 0.192259357732796$$
$$Approx := 0.192268706370918$$
$$AbsoluteError := 0.000009348638122 \quad \textbf{(33)}$$

```
> f:= x -> 2/(x^2-4);
  a:=0.0;
  b:=0.35;
  u:=t -> (b-a)/2*t+(a+b)/2;
  x1:=u(-1/sqrt(3.0)); x2:=u(1/sqrt(3.0));
  w1:=(b-a)/2; w2:=(b-a)/2;
  TrueValue:= int(f(x),x=a..b);
  Approx:= w1*f(x1) + w2*f(x2);
  AbsoluteError:=abs(TrueValue-Approx);
```

$$f := x \to \frac{2}{x^2 - 4}$$

$$a := 0.$$
$$b := 0.35$$
$$u := t \rightarrow \frac{1}{2}(b-a)\,t + \frac{1}{2}\,a + \frac{1}{2}\,b$$
$$x1 := 0.073963702891816$$
$$x2 := 0.276036297108184$$
$$w1 := 0.175000000000000$$
$$w2 := 0.175000000000000$$
$$TrueValue := -0.176820020121789$$
$$Approx := -0.176818989454912$$
$$AbsoluteError := 0.000001030666877 \quad (34)$$

Problems 3a and 3c

```
> f:= x -> x^2*ln(x);
  a:=1.0;
  b:=1.5;
  u:=t -> (b-a)/2*t+(a+b)/2;
  x1:=u(-0.7745966692); x2:=u(0); x3:=u(0.7745966692);
  w1:=(b-a)/2*10/18.0; w2:=(b-a)/2*16.0/18.0; w3:=(b-a)/2*
  10.0/18.0;
  TrueValue:= int(f(x),x=a..b);
  Approx:= w1*f(x1) + w2*f(x2) +w3*f(x3);
  AbsoluteError:=abs(TrueValue-Approx);
```

$$f := x \rightarrow x^2 \ln(x)$$
$$a := 1.0$$
$$b := 1.5$$
$$u := t \rightarrow \frac{1}{2}(b-a)\,t + \frac{1}{2}\,a + \frac{1}{2}\,b$$
$$x1 := 1.05635083270000$$
$$x2 := 1.25000000000000$$
$$x3 := 1.44364916730000$$
$$w1 := 0.138888888888889$$
$$w2 := 0.222222222222222$$
$$w3 := 0.138888888888889$$
$$TrueValue := 0.192259357732796$$
$$Approx := 0.192259377254961$$
$$AbsoluteError := 1.9522165\ 10^{-8} \quad (35)$$

```
> f:= x -> 2/(x^2-4);
  a:=0.0;
  b:=0.35;
  u:=t -> (b-a)/2*t+(a+b)/2;
  x1:=u(-0.7745966692); x2:=u(0); x3:=u(0.7745966692);
  w1:=(b-a)/2*10/18.0; w2:=(b-a)/2*16.0/18.0; w3:=(b-a)/2*
  10.0/18.0;
  TrueValue:= int(f(x),x=a..b);
  Approx:= w1*f(x1) + w2*f(x2) +w3*f(x3);
  AbsoluteError:=abs(TrueValue-Approx);
```

$$f := x \to \frac{2}{x^2 - 4}$$
$$a := 0.$$
$$b := 0.35$$
$$u := t \to \frac{1}{2}\,(b - a)\,t + \frac{1}{2}\,a + \frac{1}{2}\,b$$
$$x1 := 0.0394455828900000$$
$$x2 := 0.175000000000000$$
$$x3 := 0.310554417110000$$
$$w1 := 0.0972222222222223$$
$$w2 := 0.155555555555555$$
$$w3 := 0.0972222222222223$$
$$TrueValue := -0.176820020121789$$
$$Approx := -0.176820017886170$$
$$AbsoluteError := 2.235619\ 10^{-9} \qquad \textbf{(36)}$$

Problem 11
We need to get the correct values for 1, x, x^2, x^3
We get 4 linear equations gibing a matrix equation
L*u=b

```
> with(Student[LinearAlgebra]);
```

[&x, `.`, AddRow, AddRows, Adjoint, ApplyLinearTransformPlot, BackwardSubstitute, BandMatrix, Basis, BilinearForm, CharacteristicMatrix, CharacteristicPolynomial, ColumnDimension, ColumnSpace, CompanionMatrix, ConstantMatrix, ConstantVector, CrossProductPlot, Determinant, Diagonal, DiagonalMatrix, Dimension, Dimensions, EigenPlot, EigenPlotTutor, Eigenvalues, EigenvaluesTutor, Eigenvectors, EigenvectorsTutor, Equal, GaussJordanEliminationTutor, GaussianElimination, GaussianEliminationTutor, GenerateEquations, GenerateMatrix, GramSchmidt, HermitianTranspose, HouseholderMatrix, Id, IdentityMatrix, IntersectionBasis, InverseTutor, IsDefinite, IsOrthogonal, IsSimilar, IsUnitary, JordanBlockMatrix, JordanForm, LUDecomposition, LeastSquares, LeastSquaresPlot, LinearSolve, LinearSolveTutor, LinearSystemPlot, LinearSystemPlotTutor, LinearTransformPlot, LinearTransformPlotTutor, MatrixBuilder, MinimalPolynomial, Minor, MultiplyRow, Norm, Normalize, NullSpace, Pivot, PlanePlot, ProjectionMatrix, ProjectionPlot, QRDecomposition, RandomMatrix, RandomVector, Rank, ReducedRowEchelonForm, ReflectionMatrix, RotationMatrix, RowDimension, RowSpace, SetDefault, SetDefaults, SumBasis, SwapRow, SwapRows, Trace, Transpose, UnitVector, VectorAngle, VectorSumPlot, ZeroMatrix, ZeroVector] **(37)**

```
> A:= Matrix(4,4);
```

$$A := \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad \textbf{(38)}$$

```
> b:= Vector(4);
```

$$b := \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (39)$$

```
> f1:=  x -> 1:
  df1:= x ->0:

  for j from 2 to 4 do
  f||j := unapply(x^(j-1),x);
  df||j:= unapply((j-1)*x^(j-2),x);
  od:

  b[1]:=2;
  for j from 2 to 4 do
  b[j]:=int(f||j(x),x=-1..1);
  od:
```

$$b_1 := 2 \quad (40)$$

```
> b;
```

$$\begin{bmatrix} 2 \\ 0 \\ \frac{2}{3} \\ 0 \end{bmatrix} \quad (41)$$

```
> for j from 1 to 4
  do
  A[j,1]:= f||j(-1);
  od:

  for j from 1 to 4
  do
  A[j,2]:= f||j(1);
  od:

  for j from 1 to 4
  do
  A[j,3]:= df||j(-1);
  od:

  for j from 1 to 4
  do
  A[j,4]:= df||j(1);
  od:

> A;
```

(42)

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ -1 & 1 & 1 & 1 \\ 1 & 1 & -2 & 2 \\ -1 & 1 & 3 & 3 \end{bmatrix} \quad (42)$$

```
> u:= LinearSolve(A,b);
```

$$u := \begin{bmatrix} 1 \\ 1 \\ \frac{1}{3} \\ -\frac{1}{3} \end{bmatrix} \quad (43)$$

So the weights are 1, 1, 1/3, -1/3