```
> restart;
```

Problem 1

```
> f:= p -> (1+2^p+sqrt(2)^p+sqrt(2)^p)^(1/p);
```

$$f := p \to \left(1 + 2^p + 2\left(\sqrt{2}\right)^p\right)^{\frac{1}{p}} \tag{1}$$

```
> evalf(f(1));#1-norm
```

$$5.828427124 \tag{2}$$

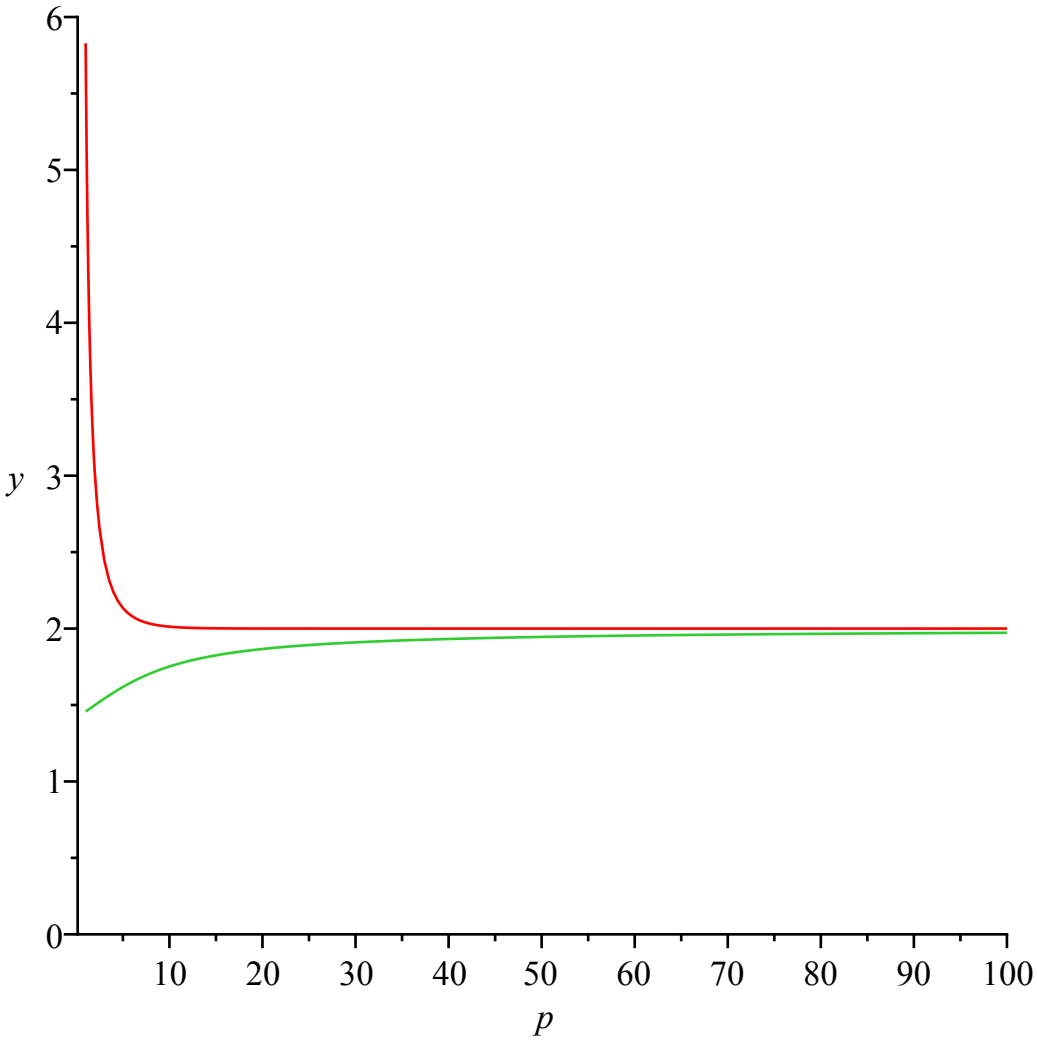
```
> simplify(f(2));#2-norm
```

$$3 \tag{3}$$

```
> limit(f(p),p=infinity);#infinity-norm
```

$$2 \tag{4}$$

```
> plot({f(p),f(p)/(4^(1/p))},p=1..100, y=0..6);
```



Problem 2

```
> restart;
> f := x -> cos(2*x)-sin(2*x);
```

$$f := x \to \cos(2x) - \sin(2x) \tag{5}$$

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

$$0 \tag{6}$$

```
> f(0); f(3/4*Pi);
```

$$1$$
$$1 \tag{7}$$

```
> with(LinearAlgebra):
> h:=evalf(3/4*Pi)/40;
```

(8)

$$h := 0.05890486225 \tag{8}$$

```
> xx := Vector(39, i-> i*h);
```

$$xx := \begin{bmatrix} 1 .. 39\ Vector_{column} \\ Data\ Type: anything \\ Storage: rectangular \\ Order: Fortran\_order \end{bmatrix} \tag{9}$$

```
> y:= Vector(39);
  r:=Vector(39);
```

$$y := \begin{bmatrix} 1 .. 39\ Vector_{column} \\ Data\ Type: anything \\ Storage: rectangular \\ Order: Fortran\_order \end{bmatrix}$$

$$r := \begin{bmatrix} 1 .. 39\ Vector_{column} \\ Data\ Type: anything \\ Storage: rectangular \\ Order: Fortran\_order \end{bmatrix} \tag{10}$$

```
> r(1):= -1;
  r(39):= -1:
  for j from 2 to 38 do
  r(j):= 0: od:
```

$$r := \begin{bmatrix} 1 .. 39\ Vector_{column} \\ Data\ Type: anything \\ Storage: rectangular \\ Order: Fortran\_order \end{bmatrix} \tag{11}$$

```
> A:= Matrix(39):
> for j from 1 to 39 do
  A(j,j):= -(2.0-4.0*h^2): od:
  for j from 1 to 38 do
  A(j,j+1):=1.0:
  A(j+1,j):=1.0:
  od:
> y:=LinearSolve(A,r);
```

$$y := \begin{bmatrix} 1 .. 39\ Vector_{column} \\ Data\ Type: float_{8} \\ Storage: rectangular \\ Order: Fortran\_order \end{bmatrix} \tag{12}$$

```
> ff:= Vector(39, i-> evalf(f(xx(i))));
```

$$ff := \begin{bmatrix} 1 .. 39\ Vector_{column} \\ Data\ Type: anything \\ Storage: rectangular \\ Order: Fortran\_order \end{bmatrix} \tag{13}$$

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

$$0.001862810849 \tag{14}$$

Problem 3

```
> restart;
> f := x -> 1/(x+4);
```

$$f := x \to \frac{1}{x+4} \tag{15}$$

```
> simplify(diff(f(x),x$2)-2*f(x)^3);
```

$$0 \tag{16}$$

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

$$\frac{1}{5}$$

$$\frac{1}{6} \tag{17}$$

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

```
> h:=0.04;
  yy:= Vector(25):
  ff:= Vector(25):
```

$$h := 0.04 \tag{18}$$

```
> dsys := t-> {diff(y(x), x$2) -2*y(x)^3 = 0, diff(yt(x), x$2) -6*yt(x)*y(x)
  ^2 = 0, y(1) = 0.2, D(y)(1)=t,yt(1)=0,D(yt)(1)=1};
```

$$dsys := t \to \left\{ \frac{d^2}{dx^2} y(x) - 2\,y(x)^3 = 0,\ \frac{d^2}{dx^2} yt(x) - 6\,yt(x)\,y(x)^2 = 0,\ y(1) = 0.2,\ yt(1) = 0,\ \mathrm{D}(y)(1) = t,\ \mathrm{D}(yt)(1) = 1 \right\} \tag{19}$$

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

$$F := t \to dsolve\left(dsys(t), numeric, method = rkf45, relerr = 1.\ \frac{1}{100000000}, output = listprocedure \right) \tag{20}$$

```
> g :=  t-> eval(y(x), F(t))-1/6.0;
  gt := t -> eval(yt(x),F(t));
```

$$g := t \to eval(y(x), F(t)) - \frac{1}{6.0}$$

$$gt := t \to eval(yt(x), F(t)) \tag{21}$$

```
> g(0.3)(2.0);gt(0.3)(2.0);#just testing syntax
```

$$0.366679016979404$$

$$1.13773614017161 \tag{22}$$

```
> 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);
```

$$t := 0.3$$

$$t := -0.0222882740844428$$

$$t := -0.0399708403103687$$

$$t := -0.0400000003028239$$

$$t := -0.0400000003785456$$

$$t := -0.0400000003785456$$
$$t := -0.0400000003785457$$
$$t := -0.0400000003785457$$
$$t := -0.0400000003785457$$
$$0. \quad (23)$$

relative error at x=2.0

```
> abs(g(t)(2.0))/ln(2.0);
```

$$0. \quad (24)$$

```
> for j from 1 to 25 do
  z:= eval(y(x), F(t))(1.0+j*h);
  yy(j):=z;
  ff(j):= 1/(5+j*h);
  od:
> Norm(yy-ff,1)/Norm(ff,1);
```

$$4.638107880\ 10^{-10} \quad (25)$$

```
> yy-ff;
```

$$\begin{bmatrix} 1 .. 25\ Vector_{column} \\ Data\ Type: float_8 \\ Storage: rectangular \\ Order: Fortran\_order \end{bmatrix} \quad (26)$$

```
> abs(yy(25)-1/6)/(1/6);
```

$$2.00000016548074\ 10^{-10} \quad (27)$$

Problem 4

```
> restart;
> with(LinearAlgebra);
```

[*&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*] (28)

```
> A:= Matrix(
  [
  [0.1,0.1+I,0.6,0.3],
```

```
[0.1,I,0.2,0.2],
[0,0.1,0.4,0.1]
]);
```

$$A := \begin{bmatrix} 0.1 & 0.1 + 1.\,I & 0.6 & 0.3 \\ 0.1 & I & 0.2 & 0.2 \\ 0 & 0.1 & 0.4 & 0.1 \end{bmatrix} \tag{29}$$

```
> Rank(A);
```

$$2 \tag{30}$$

```
> U, S, Vt := SingularValues(A, output = ['U', 'S', 'Vt']):
> U;
  S;
  Vt;
```

$$\begin{bmatrix} 0.756991080934810 + 0.\,I & -0.305992107826027 + 0.\,I & -0.577350269189626 - 0.\,I \\ 0.637315265316810 + 0.0568834560108584\,I & 0.487295699599777 + 0.140723462310503\,I & 0.577350269189626 - 4.16333634234434\,10^{-17}\,I \\ 0.119675815617999 - 0.0568834560108584\,I & -0.793287807425803 - 0.140723462310503\,I & 0.577350269189626 + 6.24500451351651\,10^{-17}\,I \end{bmatrix}$$

$$\begin{bmatrix} 1.59096298193551 \\ 0.456986641063918 \\ 2.47732774719990\,10^{-16} \end{bmatrix}$$

$$\begin{bmatrix} 0.0876391444730761 - 0.00357541040594523\,I & 0.0908570138384269 + 0.879966855136708\,I & 0.395689928062052 + 0.00715082081189042\,I & 0.230381198725127 - 0.00357541040594522\,I \\ 0.0396737181094937 - 0.0307937803133333\,I & 0.0673881203914950 + 0.427530961408285\,I & -0.882851294748357 + 0.0615875606266663\,I & -0.161202246522846 - 0.0307937803133332\,I \\ -0.991831353405041 + 0.0755064533390510\,I & 0.00405692171090484 + 0.0958125457277876\,I & -0.00696995659149363 - 0.0261146425375961\,I & 0.0238229046550752 + 0.00864602442259532\,I \\ 0.00862060863211200 - 0.0167141555435301\,I & 0.00596126682764585 - 0.144386342601326\,I & -0.241194225172740 + 0.0354076435637772\,I & 0.958815633863312 + 0.00275576834621709\,I \end{bmatrix} \tag{31}$$

```
> Sigma:= Matrix(3,4);
```

$$\Sigma := \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{32}$$

```
> for j from 1 to 3 do Sigma(j,j):=S(j); od:
  Sigma;
```

$$\begin{bmatrix} 1.59096298193551 & 0 & 0 & 0 \\ 0 & 0.456986641063918 & 0 & 0 \\ 0 & 0 & 2.47732774719990\,10^{-16} & 0 \end{bmatrix} \tag{33}$$

```
> U . Sigma .Vt-A;
```

$$\begin{bmatrix} 1.94289029309402\,10^{-16} - 1.07995811169094\,10^{-17}\,I & 1.38777878078145\,10^{-17} + 0.\,I & 0. + 3.73514034830187\,10^{-18}\,I & 0. - 1.23662863187757\,10^{-18}\,I \\ -5.82867087928207\,10^{-16} - 1.31838984174237\,10^{-16}\,I & -1.38777878078145\,10^{-17} + 0.\,I & -2.77555756156289\,10^{-16} - 5.20417042793042\,10^{-17}\,I & -8.32667268468867\,10^{-17} \end{bmatrix} \tag{34}$$

$-2.60208521396521\ 10^{-17}\ I],$
$[3.43862339058903\ 10^{-16} + 1.44373288767124\ 10^{-16}\ I,\ 4.16333634234434\ 10^{-17}$
$-6.95627940855570\ 10^{-17}\ I,\ -5.55111512312578\ 10^{-17} + 6.56537986907704\ 10^{-17}\ I,$
$-2.77555756156289\ 10^{-17} + 2.55227572955529\ 10^{-17}\ I]]$

The first two columns of U are orthonormal and span the range of A

```
> V:= HermitianTranspose(Vt);
```

$$V := [[0.0876391444730761 + 0.00357541040594523\ I,\ 0.0396737181094937 + 0.0307937803133333\ I,\ -0.991831353405041 - 0.0755064533390510\ I,\ 0.00862060863211200 + 0.0167141555435301\ I],\ [0.0908570138384269 - 0.879966855136708\ I,\ 0.0673881203914950 - 0.427530961408285\ I,\ 0.00405692171090484 - 0.0958125457277876\ I,\ 0.00596126682764585 + 0.144386342601326\ I],\ [0.395689928062052 - 0.00715082081189042\ I,\ -0.882851294748357 - 0.0615875606266663\ I,\ -0.00696995659149363 + 0.0261146425375961\ I,\ -0.241194225172740 - 0.0354076435637772\ I],\ [0.230381198725127 + 0.00357541040594522\ I,\ -0.161202246522846 + 0.0307937803133332\ I,\ 0.0238229046550752 - 0.00864602442259532\ I,\ 0.958815633863312 - 0.00275576834621709\ I]] \quad (35)$$

```
> e1:= Vector(4):
  e2:=Vector(4):
> for j from 1 to 4 do
  e1(j):= V(j,3):
  e2(j):= V(j,4):
  od:
> A . e1;#checking they are truly the nullspace
```

$$\begin{bmatrix} 4.16333634234434\ 10^{-16} + 3.42607886505419\ 10^{-17}\ I \\ -1.43982048506075\ 10^{-16} - 1.02565525517129\ 10^{-16}\ I \\ 5.56846235788555\ 10^{-16} + 1.37368415253913\ 10^{-16}\ I \end{bmatrix} \quad (36)$$

```
> A . e2;#checking they are truly the nullspace
```

$$\begin{bmatrix} 2.77555756156289\ 10^{-17} - 6.72205346941013\ 10^{-18}\ I \\ 0. + 4.33680868994202\ 10^{-19}\ I \\ 1.38777878078145\ 10^{-17} - 9.75781955236954\ 10^{-18}\ I \end{bmatrix} \quad (37)$$

Problem 5

```
> restart;
> with(LinearAlgebra);
```

[*&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*, (38)

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*]

```
> Digits:=20;
```

$$Digits := 20 \tag{39}$$

```
> N:=11:
  v:=Vector(N):

> for j from 1 to N do v[j]:=j-1: od:
> V:= evalf(Matrix(N,shape=Vandermonde[v]));#Matrix(N,shape=Vandermonde[v]);#
```

$$V := \begin{bmatrix} 11 \text{ x } 11 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix} \tag{40}$$

```
> evalf(ConditionNumber(V,2));
```

$$4.4628225179975030898\ 10^{12} \tag{41}$$

```
> e,T := Eigenvectors(V):
> e;
  DiagonalMatrix(e);
```

$$\begin{bmatrix} 1 .. 11 \text{ Vector}_{column} \\ \text{Data Type: complex(sfloat)} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix}$$

$$\begin{bmatrix} 11 \text{ x } 11 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: diagonal} \\ \text{Order: Fortran\_order} \end{bmatrix} \tag{42}$$

```
> Norm(MatrixInverse(T) . V . T - DiagonalMatrix(e));
```

$$3.7950611\ 10^{-9} \tag{43}$$

```
> ConditionNumber(evalf(T));
```

$$17.515860587456370106 \tag{44}$$

Problem 6

```
> restart;
> with(LinearAlgebra):
> Digits:=20;
```

$$Digits := 20 \tag{45}$$

```
> N:=11:
  v:=Vector(N):

> for j from 1 to N do v[j]:=exp(2.0*Pi*(j-1)*I/11): od:
> A:= evalf(Matrix(N,shape=Vandermonde[v]));
```

$$A := \begin{bmatrix} 11 \text{ x } 11 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix} \quad (46)$$

```
> ConditionNumber(A,2);
```

$$1.0000000000000000006 \quad (47)$$

```
> e,g := Eigenvectors(A):
> e;
  DiagonalMatrix(e);
```

$$\begin{bmatrix} 1 .. 11 \text{ Vector}_{column} \\ \text{Data Type: complex(sfloat)} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix}$$

$$\begin{bmatrix} 11 \text{ x } 11 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: diagonal} \\ \text{Order: Fortran\_order} \end{bmatrix} \quad (48)$$

```
> Norm(MatrixInverse(g) . A . g - DiagonalMatrix(e));
```

$$6.4899789581826680226\ 10^{-18} \quad (49)$$

```
> ConditionNumber(g);
```

$$20.536185756285849237 \quad (50)$$