

```

% Angewandte Numerik 1, Blatt 2, Aufgabe 6:
%
% Berechnung der 1-Norm (=Spaltensummennorm), der Unendlich-Norm
% (=Zeilensummennorm) und der Frobenius-Norm fuer zufaellig erzeugte
% Matrizen mit unterschiedlichen Dimensionen und unterschiedlichen
% Wertebereichen.
%
% Letzte Aenderung: 06.05.2020

% Initialisierung
clearvars;
fprintf( "\nAngewandte Numerik 1, Blatt 2, Aufgabe 6\n" );

% Spezifikation der Variante der mynorm Function
mynorm = @mynorm3;

% Es werden 4 verschiedene Matrizen erzeugt
% Anzahl der Zeilen der Matrizen
rows = [ 4, 5, 3, 7 ];
% Anzahl der Spalten der Matrizen
cols = [ 4, 3, 6, 5 ];
% Verschiebung des Wertebereichs der rand Funktion
% mit shift = -0.5 Verschiebung des Wertebereichs von [0,1] nach [-0.5,0.5]
shift = [ 0, -0.5, -0.5, 0 ];
% Skalierung des Wertebereichs der rand Funktion
scal = [ 50, 200, 30, 100 ];

% Berechnung der Normen der Matrizen
for k = 1:length(rows)
    % Erzeugung der Matrix
    A = ( rand( rows(k), cols(k) ) + shift(k) ) * scal(k);

    % Ausgabe der Matrix
    fprintf( "\nMatrix A:\n" );
    disp( A )

    % Ausgabe der Spaltensummennorm der Matrix
    fprintf( "Spaltensummennorm von A = %8.4f", mynorm( A, ('one')) ) );
    fprintf( ", Kontrolle mit Matlab norm(A,1) = %8.4f\n", norm(A, 1) );

    % Ausgabe der Zeilensummennorm der Matrix
    fprintf( "Zeilensummennorm von A = %8.4f", mynorm( A, 'infty' ) );
    fprintf( ", Kontrolle mit Matlab norm(A,Inf) = %8.4f\n", ...
        norm(A, Inf) );

    % Ausgabe der Frobenius-Norm der Matrix

```

```

fprintf( "Frobenius-Norm von A = %8.4f", mynorm( A, 'frobenius' ) );
fprintf( ", Kontrolle mit Matlab norm(A,'fro') = %8.4f\n\n", ...
        norm(A, 'fro') );
end

#####

function Norm = mynorm1( A, flag )
% Angewandte Numerik 1, Blatt 2, Aufgabe 6:
% Die Function mynorm gibt in Abhaengigkeit vom Eingabeparameter flag
% die 1-Norm (=Spaltensummennorm), die Unendlich-Norm (=Zeilensummennorm)
% oder die Frobenius-Norm der uebergebenen Matrix A zurueck
%
% Einfache Variante mit Character Array, if/else Abfrage und for Loop
%
% Letzte Aenderung: 06.05.2020
%
% Input: A Matrix A, von der die Matrix-Normen berechnet werden sollen
% flag Flag, das festlegt, ob die Spaltensummennorm, die
% Zeilensummennorm oder die Frobenius-Norm der Matrix A
% berechnet werden soll (Character-Vektor oder String)
% Output: Norm Die berechnete Norm (mit grossem N am Anfang um die Matlab
% Funktion norm nicht zu ueberschreiben)

% Initialisierung und Fehlerbehandlung
% Sicherstellen, dass flag ein Character-Vektor oder ein String ist
if( ~ischar(flag) && ~isstring(flag) )
    % Fehler: flag ist kein String
    fprintf( "\nEingabevariable flag = %s ist kein String !!\n\n", ...
            string( flag ) );
    % Die Ausgabevariable muss auch im Fehlerfall gesetzt sein
    Norm = -1;
    return;
end

% Sicherstellen, dass flag ein Zeilenvektor ist (fuer Vergleich mit strcmp)
flag = flag(:)';

% Groesse der Matrix bestimmen
[m,n] = size( A );

% Initialisierung der Arrays
% Spaltensummen
ssum = zeros( 1, n );
% Zeilensummen
zsum = zeros( 1, m );

```

```

% Berechnung der Normen
if strcmp( flag, 'one' )
    % Berechnung der 1-Norm = Spaltensummennorm
    for ks = 1:n
        for kz = 1:m
            ssum(ks) = ssum(ks) + abs( A(kz,ks) );
        end
    end
    Norm = max( ssum );
elseif strcmp( flag, 'infty' )
    % Berechnung der Unendlich-Norm = Zeilensummennorm
    for kz = 1:m
        for ks = 1:n
            zsum(kz) = zsum(kz) + abs( A(kz,ks) );
        end
    end
    Norm = max( zsum );
elseif strcmp( flag, 'frobenius' )
    % Berechnung der Frobenius-Norm
    fsum = 0;
    for ks = 1:n
        for kz = 1:m
            fsum = fsum + ( A(kz,ks) )^2;
        end
    end
    Norm = sqrt( fsum );
else
    % Keine bekannte Norm
    fprintf( "\nNorm %s ist nicht bekannt !!\n\n", flag );
    % Die Ausgabevariable muss auch im Fehlerfall gesetzt sein
    Norm = -1;
    return;
end
end

#####

function Norm = mynorm2( A, flag )
% Angewandte Numerik 1, Blatt 2, Aufgabe 6:
% Die Function mynorm gibt in Abhaengigkeit vom Eingabeparameter flag
% die 1-Norm (=Spaltensummennorm), die Unendlich-Norm (=Zeilensummennorm)
% oder die Frobenius-Norm der uebergebenen Matrix A zurueck
%
% Verbesserte Variante mit String Array, if/else Abfrage und sum Funktion
%
```

```

% Bei der Matlab Funktion sum(M,d) gibt der 2. Parameter die Richtung der
% Summation in der Matrix an: d=1: Summation der Spalten der Matrix M
%                               d=2: Summation der Zeilen der Matrix M
%                               d=[1 2]: Summation aller Elemente der Matrix
%
% Letzte Aenderung: 06.05.2020
%
% Input:  A      Matrix A, von der die Matrix-Normen berechnet werden sollen
%         flag    Flag, das festlegt, ob die Spaltensummenorm, die
%                Zeilensummenorm oder die Frobenius-Norm der Matrix A
%                berechnet werden soll (Character-Vektor oder String)
% Output: Norm    Die berechnete Norm (mit grossem N am Anfang um die Matlab
%                Funktion norm nicht zu ueberschreiben)

% Initialisierung und Fehlerbehandlung
% Sicherstellen, dass flag ein Character-Vektor oder ein String ist
if( ~ischar(flag) && ~isstring(flag) )
    % Fehler: flag ist kein String
    fprintf( "\nEingabevariable flag = %s ist kein String !!\n\n", ...
            string( flag ) );
    % Die Ausgabevariable muss auch im Fehlerfall gesetzt sein
    Norm = -1;
    return;
end

% Sicherstellen, dass flag ein Zeilenvektor ist, Umwandeln in String
flag = string( flag(:)' );

% Berechnung der Normen
if( flag == "one" )
    % Berechnung der 1-Norm = Spaltensummenorm
    Norm = max( sum( abs(A), 1 ) );
elseif( flag == "inf" )
    % Berechnung der Unendlich-Norm = Zeilensummenorm
    Norm = max( sum( abs(A), 2 ) );
elseif( flag == "frobenius" )
    % Berechnung der Frobenius-Norm
    % (zunaechst Matrix elementweise quadrieren)
    Norm = sqrt( sum( A.^2, [1 2] ) );
else
    % Keine bekannte Norm
    fprintf( "\nNorm %s ist nicht bekannt !!\n\n", flag );
    % Die Ausgabevariable muss auch im Fehlerfall gesetzt sein
    Norm = -1;
    return;
end

```

end

#####

```
function Norm = mynorm3( A, flag )
% Angewandte Numerik 1, Blatt 2, Aufgabe 6:
% Die Function mynorm gibt in Abhaengigkeit vom Eingabeparameter flag
% die 1-Norm (=Spaltensummennorm), die Unendlich-Norm (=Zeilensummennorm)
% oder die Frobenius-Norm der uebergebenen Matrix A zurueck
%
% Verbesserte Variante mit switch/case und sum Funktion
%
% Bei der Matlab Funktion sum(M,d) gibt der 2. Parameter die Richtung der
% Summation in der Matrix an: d=1: Summation der Spalten der Matrix
%                               d=2: Summation der Zeilen der Matrix
%                               d=[1 2]: Summation aller Elemente der Matrix
% Die switch/case Funktion verarbeitet Character-Zeilenvektoren,
% Strings (mit "abc") richtig, so dass dafuer keine Anpassungen noetig sind
%
% Letzte Aenderung: 06.05.2020
%
% Input:  A      Matrix A, von der die Matrix-Normen berechnet werden sollen
%         flag   Flag, das festlegt, ob die Spaltensummennorm, die
%                Zeilensummennorm oder die Frobenius-Norm der Matrix A
%                berechnet werden soll (Character-Vektor oder String)
% Output: Norm   Die berechnete Norm (mit grossem N am Anfang um die Matlab
%                Funktion norm nicht zu ueberschreiben)

% Initialisierung und Fehlerbehandlung
% Sicherstellen, dass flag ein Character-Vektor oder ein String ist
if( ~ischar(flag) && ~isstring(flag) )
    % Fehler: flag ist kein String
    fprintf( "\nEingabevariable flag = %s ist kein String !!\n\n", ...
            string( flag ) );
    % Die Ausgabevariable muss auch im Fehlerfall gesetzt sein
    Norm = -1;
    return;
end

% Sicherstellen, dass flag ein Zeilenvektor ist
flag = flag(:)';

% Berechnung der Normen
switch flag
case 'one'
    % Berechnung der 1-Norm = Spaltensummennorm
```

```

        Norm = max( sum( abs(A), 1 ) );
    case "infty"
        % Berechnung der Unendlich-Norm = Zeilensummennorm
        Norm = max( sum( abs(A), 2 ) );
    case 'frobenius'
        % Berechnung der Frobenius-Norm
        % (zunächst Matrix elementweise quadrieren)
        Norm = sqrt( sum( A.^2, [1 2] ) );
    otherwise
        % Keine bekannte Norm
        fprintf( "\nNorm %s ist nicht bekannt !!\n\n", flag );
        % Die Ausgabevariable muss auch im Fehlerfall gesetzt sein
        Norm = -1;
        return;
    end
end
end

```

Angewandte Numerik 1, Blatt 2, Aufgabe 6

Matrix A:

Columns 1 through 3

22.075247580836937	18.851306140722318	1.065826224093364
45.722916860096483	5.646842609439911	21.724830712734612
31.235339420897617	36.210033211024786	29.737626491453291
6.657821732446090	33.226202093635223	25.007538075068997

Column 4

9.684867434205513
29.064696268803448
22.228747551378763
36.552592673773262

Spaltensummennorm von A = 105.6913, Kontrolle mit Matlab `norm(A,1)` = 105.6913  
 Zeilensummennorm von A = 119.4117, Kontrolle mit Matlab `norm(A,Inf)` = 119.4117  
 Frobenius-Norm von A = 105.6023, Kontrolle mit Matlab `norm(A,'fro')` = 105.6023

Matrix A:

-45.531775654631886	-22.429058910351163	81.296188498227224
-89.195573988786435	-75.569005617054572	-90.132290925768288
-17.163047257704235	59.109122812909852	81.904357930840391
41.428522363964589	-62.349460106911380	-15.219094682994339
69.647113151187085	47.063420780986753	-71.773250898265118

Spaltensummennorm von A = 340.3252, Kontrolle mit Matlab norm(A,1) = 340.3252  
 Zeilensummennorm von A = 254.8969, Kontrolle mit Matlab norm(A,Inf) = 254.8969  
 Frobenius-Norm von A = 243.9772, Kontrolle mit Matlab norm(A,'fro') = 243.9772

Matrix A:

Columns 1 through 3

```
-2.685893379691948  11.274078452734264 -12.016809882998098
-2.268225088151188  12.701836517183473  14.217533566160684
-5.115421167018896 -13.049792668155643 -12.552185745437773
```

Columns 4 through 6

```
-9.000144177205357 -11.498088772721131 -8.011928377952703
-9.929853898187064 -13.860391914044863 -10.710247379481107
 4.140462770395875 -9.908327493420140 -14.570206402858316
```

Spaltensummennorm von A = 38.7865, Kontrolle mit Matlab norm(A,1) = 38.7865  
 Zeilensummennorm von A = 63.6881, Kontrolle mit Matlab norm(A,Inf) = 63.6881  
 Frobenius-Norm von A = 44.8530, Kontrolle mit Matlab norm(A,'fro') = 44.8530

Matrix A:

Columns 1 through 3

```
79.626460249758892  55.317843422878667  59.597511614219002
47.812823672848239   8.550078036838848  76.624507800092204
82.887994274018340  52.720623938329716  43.989024685895075
 4.640543144090092  84.408060441149843  95.673243384683872
99.043026514195631  49.381143593950547  46.247941229607434
86.292813725289690  24.339725182573790  55.215838448023561
74.191155517742004  64.540797685807433  48.216687361806429
```

Columns 4 through 5

```
62.237435385080467  54.557219885196197
50.853816396179411  82.064790446371646
37.874565648535842  54.157327256998876
70.626200555964431  79.973621397599331
73.339599818783157   1.473740813741298
24.540567027564244  18.602690249141361
34.674219546550809  56.159726843321501
```

Spaltensummennorm von A = 474.4948, Kontrolle mit Matlab norm(A,1) = 474.4948

Zeilensummennorm von A = 335.3217, Kontrolle mit Matlab `norm(A,Inf)` = 335.3217  
Frobenius-Norm von A = 359.3661, Kontrolle mit Matlab `norm(A,'fro')` = 359.3661