Gruppe 1
Freitag 13:15-14:45, B-Woche
(Dr. B. Schulze)

Claudia Heinich, 8966108
Sven Pohl, 8968075
Arne Brutschy, 8964813

Numerik

Serie 9

17. Wir berechnen die Cholesky-Zerlegung:

$$\begin{aligned} A &= L \cdot L^T \\ &= \begin{pmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{pmatrix} \cdot \begin{pmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{pmatrix} \\ &= \begin{pmatrix} l_{11}^2 & l_{11}l_{21} & l_{11}l_{31} \\ l_{11}l_{21} & l_{21}^2 + l_{22}^2 & l_{21}l_{31} + l_{22}l_{23} \\ l_{11}l_{31} & l_{21}l_{31} + l_{22}l_{23} & l_{31}^2 + l_{32}^2 + l_{33}^2 \end{pmatrix} \end{aligned}$$

Woraus folgt:

$$\begin{aligned} l_{11}^2 &= a_{11} \\ l_{11} &= \sqrt{a_{11}} = 3 \end{aligned}$$

$$\begin{aligned} l_{21}l_{11} &= a_{12} = a_{21} \\ l_{21} &= \frac{a_{12}}{l_{11}} = \frac{-6}{3} = -2 \end{aligned}$$

$$\begin{aligned} l_{31}l_{11} &= a_{13} = a_{31} \\ l_{31} &= \frac{a_{13}}{l_{11}} = \frac{3}{3} = 1 \end{aligned}$$

$$\begin{aligned} l_{22}^2 + l_{21}^2 &= a_{22} \\ l_{22}^2 &= a_{22} - l_{21}^2 \\ l_{22} &= \sqrt{a_{22} - l_{21}^2} = \sqrt{5-4} = 1 \end{aligned}$$

$$\begin{aligned} l_{32}l_{22} + l_{21}l_{31} &= a_{32} = a_{23} \\ l_{32}l_{22} &= a_{32} - l_{21}l_{31} \\ l_{32} &= \frac{a_{3,2} - l_{21}l_{31}}{l_{22}} = \frac{-4+2}{1} = -2 \end{aligned}$$

$$\begin{aligned} l_{33}^2 + l_{32}^2 + l_{31}^2 &= a_{33} \\ l_{33}^2 &= a_{33} - l_{32}^2 - l_{31}^2 \\ l_{33} &= \sqrt{a_{33} - l_{32}^2 - l_{31}^2} = \sqrt{9-4-1} = 2 \end{aligned}$$

Somit ist

$$L = \begin{pmatrix} 3 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & -2 & 2 \end{pmatrix}$$

Claudia Heinich, 8966108
Sven Pohl, 8968075
Arne Brutschy, 8964813

18. a) Hier die Ergebnisse des Vergleiches zwischen dem Gauss-Algorithmus mit Spaltenpivotsuche und der QR-Zerlegung:

n	Gauss	QR	Differenz
1	1.000000000	1.000000000	0.000000000
1	1.000000000	1.000000000	0.000000000
2	1.000000000	1.000000000	-0.000000000
1	1.000000000	1.000000000	0.000000000
2	1.000000000	1.000000000	-0.000000000
3	1.000000000	1.000000000	0.000000000
1	1.000000000	1.000000000	-0.000000000
2	1.000000000	1.000000000	0.000000000
3	1.000000000	1.000000000	-0.000000000
4	1.000000000	1.000000000	0.000000000
1	1.000000000	1.000000000	0.000000000
2	1.000000000	1.000000000	-0.000000000
3	1.000000000	1.000000000	0.000000000
4	1.000000000	1.000000000	-0.000000000
5	1.000000000	1.000000000	0.000000000
1	1.000000000	1.000000000	0.000000000
2	1.000000000	1.000000000	-0.000000000
3	1.000000000	1.000000000	0.000000000
4	1.000000000	1.000000000	-0.000000000
5	1.000000000	0.999999999	0.000000000
6	1.000000000	1.000000000	-0.000000000
1	1.000000000	1.000000000	0.000000000
2	1.000000000	1.000000000	-0.000000000
3	1.000000000	1.000000000	0.000000000
4	1.000000001	1.000000001	-0.000000000
5	0.999999998	0.999999998	0.000000000
6	1.000000002	1.000000002	0.000000000
7	0.999999999	0.999999999	-0.000000000
1	1.000000000	1.000000000	0.000000000
2	0.999999999	1.000000000	-0.000000001
3	1.000000016	1.000000006	0.000000010
4	0.999999912	0.999999968	-0.000000056
5	1.000000232	1.000000083	0.000000150
6	0.999999675	0.999999886	-0.000000211
7	1.000000229	1.000000079	0.000000150
8	0.999999936	0.999999978	-0.000000042
1	1.000000000	1.000000000	-0.000000000
2	0.999999994	0.999999993	0.000000001
3	1.000000105	1.000000121	-0.000000015
4	0.999999255	0.999999142	0.000000113
5	1.000002718	1.000003145	-0.000000427
6	0.999994472	0.999993579	0.000000892
7	1.000006334	1.000007378	-0.000001044
8	0.999996180	0.999995540	0.000000640
9	1.000000943	1.000001103	-0.000000160

Claudia Heinich, 8966108
Sven Pohl, 8968075
Arne Brutschy, 8964813

n	Gauss	QR	Differenz
1	1.000000000	1.000000000	0.000000000
2	0.999999967	0.999999994	-0.000000028
3	1.000000731	1.000000136	0.000000595
4	0.999993197	0.999998633	-0.000005436
5	1.000033097	1.000007071	0.000026026
6	0.999907484	0.999979257	-0.000071772
7	1.000153976	1.000035891	0.000118085
8	0.999849353	0.999963750	-0.000114397
9	1.000079943	1.000019754	0.000060189
10	0.999982253	0.999995515	-0.000013262
1	1.000000000	1.000000001	-0.000000000
2	0.999999943	0.999999906	0.000000038
3	1.000001615	1.000002701	-0.000001086
4	0.999980604	0.999967563	0.000013041
5	1.000122173	1.000204200	-0.000082027
6	0.999550947	0.999249941	0.000301006
7	1.001013765	1.001692242	-0.000678477
8	0.998575839	0.997624043	0.000951797
9	1.001213263	1.002023109	-0.000809847
10	0.999426419	0.999043966	0.000382453
11	1.000115432	1.000192330	-0.000076897
1	1.000000013	0.999999977	0.000000036
2	0.999998313	1.000002852	-0.000004539
3	1.000053781	0.999910914	0.000142867
4	0.999260754	1.001207677	-0.001946923
5	1.005449796	0.991180800	0.014268996
6	0.975970678	1.038632544	-0.062661866
7	1.067091752	0.892620347	0.174471406
8	0.878420513	1.193986509	-0.315565996
9	1.142597521	0.772949305	0.369648216
10	0.895569796	1.166056704	-0.270486908
11	1.043402163	0.931040307	0.112361856
12	0.992184910	1.012412080	-0.020227170
1	0.999999982	0.999999806	0.000000177
2	1.000002712	1.000030162	-0.000027450
3	0.999895357	0.998845038	0.001050319
4	1.001756127	1.019153704	-0.017397577
5	0.984054855	0.828399441	0.155655414
6	1.087558070	1.929641980	-0.842083910
7	0.690699335	-2.241457073	2.932156409
8	1.725967894	8.515317472	-6.789349577
9	-0.143718307	-10.705553264	10.561834957
10	2.195177281	13.103840123	-10.908662842
11	0.205604508	-6.967171096	7.172775604
12	1.303966234	4.021362932	-2.717396698
13	0.949035930	0.497590601	0.451445329

Claudia Heinich, 8966108
Sven Pohl, 8968075
Arne Brutschy, 8964813

n	Gauss	QR	Differenz
1	1.000000054	1.000000003	0.000000051
2	0.999991139	0.999999030	-0.000007891
3	1.000357907	1.000055404	0.000302503
4	0.993705284	0.998778175	-0.005072891
5	1.060318495	1.013875813	0.046442682
6	0.646440059	0.906577231	-0.260137172
7	2.354219345	1.403189413	0.951029932
8	-2.519260351	-0.165781521	-2.353478829
9	7.316736004	3.307988155	4.008747848
10	-6.836800803	-2.138928820	-4.697871983
11	7.598117391	3.884256203	3.713861188
12	-2.598206175	-0.711069283	-1.887136891
13	2.146337450	1.591629107	0.554708343
14	0.838044213	0.909431085	-0.071386871
1	1.000000046	1.000000034	0.000000012
2	0.999992940	0.999995188	-0.000002248
3	1.000268738	1.000169219	0.000099519
4	0.995594994	0.997452013	-0.001857019
5	1.038685534	1.020321110	0.018364424
6	0.797545843	0.905368323	-0.107822480
7	1.662758958	1.265149549	0.397609409
8	-0.355976778	0.577502132	-0.933478909
9	2.577568183	1.245711238	1.331856945
10	0.499883164	1.390814503	-0.890931338
11	-0.386922014	0.009965609	-0.396887623
12	3.375475327	1.970553487	1.404921840
13	-0.776349792	0.504756071	-1.281105863
14	1.679453821	1.121443124	0.558010698
15	0.892021051	0.990798427	-0.098777375
1	0.999999567	0.999999816	-0.000000250
2	1.000066416	1.000028391	0.000038024
3	0.997492834	0.998918207	-0.001425372
4	1.040514141	1.017685397	0.022828744
5	0.654645878	0.846599625	-0.191953748
6	2.702351109	1.780435028	0.921916081
7	-3.943969542	-1.431415537	-2.512554005
8	8.730990164	5.656857116	3.074133048
9	-2.024983624	-4.474880470	2.449896846
10	-8.945275698	6.252493737	-15.197769435
11	14.930813518	-8.850335586	23.781149104
12	6.574522382	22.311172185	-15.736649803
13	-29.026146276	-27.667226129	-1.358920147
14	32.449110748	23.195360147	9.253750601
15	-13.956778051	-8.260503639	-5.696274412
16	3.816646787	2.624811883	1.191834904

Claudia Heinich, 8966108
Sven Pohl, 8968075
Arne Brutschy, 8964813

n	Gauss	QR	Differenz
1	0.999999894	0.999999951	-0.000000056
2	1.000015301	1.000007346	0.000007955
3	0.999452223	0.999725710	-0.000273487
4	1.008389934	1.004453269	0.003936665
5	0.932913687	0.961551382	-0.028637695
6	1.299928565	1.190772920	0.109155645
7	0.295848021	0.468828533	-0.172980512
8	1.371042267	1.585029065	-0.213986798
9	3.587794013	2.098145725	1.489648287
10	-6.819811348	-4.085174225	-2.734637123
11	10.548228196	8.564842076	1.983386121
12	-1.513296686	-1.749736197	0.236439512
13	-7.209610531	-6.645264941	-0.564345590
14	13.174875405	14.671388909	-1.496513504
15	-6.944591514	-9.513813371	2.569221857
16	3.622767324	5.120377952	-1.497610628
17	0.646055279	0.328865880	0.317189399
1	0.999999763	0.999999747	0.000000016
2	1.000031868	1.000030978	0.000000890
3	0.998965607	0.999119964	-0.000154357
4	1.013870477	1.009421969	0.004448509
5	0.908348786	0.963865152	-0.055516366
6	1.298442259	0.925707226	0.372735033
7	0.715428625	2.179889288	-1.464460663
8	-0.096696504	-3.525748301	3.429051798
9	4.714339488	9.336159183	-4.621819695
10	-1.901821207	-5.513962856	3.612141648
11	-4.748411441	-0.675206082	-4.073205359
12	18.316868793	9.105914318	9.210954475
13	-23.965661783	-15.155845960	-8.809815823
14	32.156844203	40.242766519	-8.085922316
15	-34.139895122	-61.372575904	27.232680782
16	28.933192303	55.820991089	-26.887798786
17	-11.594413561	-24.018176769	12.423763208
18	3.390567740	5.677650826	-2.287083086

Claudia Heinich, 8966108
Sven Pohl, 8968075
Arne Brutschy, 8964813

n	Gauss	QR	Differenz
1	0.999999904	0.999999923	-0.000000019
2	1.000016167	1.000010560	0.000005607
3	0.999318547	0.999650760	-0.000332213
4	1.012478748	1.004665496	0.007813252
5	0.877111190	0.971962173	-0.094850983
6	1.720939373	1.045457275	0.675482098
7	-1.634453856	1.377236042	-3.011689898
8	6.964995377	-1.642368546	8.607363923
9	-6.596883420	8.849906586	-15.446790006
10	3.871867686	-11.689931735	15.561799421
11	6.705138048	11.317233190	-4.612095142
12	-6.738921308	-0.750651761	-5.988269547
13	2.350089093	-1.296736801	3.646825894
14	2.975434888	2.011948831	0.963486057
15	3.104672502	-5.124426765	8.229099267
16	-3.809826120	17.065656322	-20.875482442
17	3.559310182	-15.657373225	19.216683406
18	0.738018837	9.039421062	-8.301402225
19	0.900694105	-0.521659426	1.422353531
1	0.999998155	1.000000063	-0.000001907
2	1.000275582	0.999989986	0.000285596
3	0.989924708	1.000391628	-0.010466920
4	1.155760885	0.993398552	0.162362333
5	-0.241397848	1.060358152	-1.301756000
6	6.465782151	0.658335520	5.807446632
7	-11.669260683	2.324686735	-13.993947419
8	10.152890271	-2.849108797	13.001999068
9	24.709500256	9.875337216	14.834163040
10	-56.375607784	-14.962322637	-41.413285147
11	25.433091317	21.487565100	3.945526217
12	45.511806004	-15.990744044	61.502550048
13	-40.287354818	12.259600206	-52.546955023
14	23.063793383	-14.001378205	37.065171588
15	-99.514405839	18.730152911	-118.244558750
16	136.418226550	3.510877644	132.907348907
17	0.950537197	-30.565802276	31.516339473
18	-118.485610654	36.460152760	-154.945763414
19	89.246914016	-16.248290255	105.495204270
20	-19.524863820	4.256799827	-23.781663647

Wie man deutlich sehen kann ist der QR-Algorithmus für gerade n stabiler als der Gauss-Algorithmus. Für ungerade n scheint es im Algorithmus zu einer subtraktiven Auslöschung zu kommen, wodurch die Ergebnisse bei wachsenden n stark verfälscht werden. Ansonsten kann man beim Algorithmus einen nahezu linearen Anstieg des Fehlers beobachten.

Claudia Heinich, 8966108
Sven Pohl, 8968075
Arne Brutschy, 8964813

b) Hier der zugehörige Sourcecode. Wir haben den Algorithmus getrennt, so daß A und b von getrennten Funktionen berechnet werden kann. Dafür wird der Vektor d_i, $i = 1, \ldots, n$ in den Aufruflisten mitübergeben. Dies hat den Vorteil, daß man eine Matrix erst zerlegen kann und dann mit geringen Rechenaufwand verschiedene Ergebnisse b berechnen kann. Es ist zu beachten, daß wir den Gauss-Algorithmus zu Vergleichszwecken hinzugefügt haben. Ausserdem erzeugt das Programm seine Ausgaben direkt im LaTeX-Syntax.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>



typedef struct {
  double A[20][20];
  double b[20];
  double x[20];
  double d[20];
  int n;
} gleichung_t;



void ausgabe_ergebnis(gleichung_t *gl_ohne, gleichung_t *gl_mit) {
  int i;

  for (i= 0; i < gl_ohne->n; i++) {
    printf("          %2d & %3.9f & %3.9f & %3.9f \\\\\n", i + 1,
      gl_ohne->x[i], gl_mit->x[i], gl_ohne->x[i] - gl_mit->x[i]);
  }
  printf("          \\hline \n");
  return;
}


void hilbert(gleichung_t *gl) {
  int i, j;

  for (i= 0; i < gl->n; i++) {
    gl->b[i]= 0;

    for (j= 0; j < gl->n; j++) {
      gl->A[i][j]= 1 / (double)(i + j + 2 - 1);
      gl->b[i]+= gl->A[i][j];
    }
  }

  return;
}
```

Claudia Heinich, 8966108
Sven Pohl, 8968075
Arne Brutschy, 8964813

```
void qr_zerlegung(gleichung_t *gl) {
  long double  beta, tau, gamma;
  int     i, j, k, l;

  for (i= 0; i < gl->n; i++) {
    tau= pow(gl->A[i][i], 2.0);
    for (j= i+1; j < gl->n; j++)
      tau+= pow(gl->A[j][i], 2.0);

    gl->d[i]= sqrt(tau);
    if (gl->d[i] == 0) {
      printf("Matrix ist singulaer!\n");
      return;
    }

    //gl->d[i]= (gl->A[i][i] <= 0) ? (gl->d[i]) : (-gl->d[i]);
    if (gl->A[i][i] > 0) {
       gl->d[i] = -gl->d[i];
    }
    beta=(long double)1 / (gl->d[i] * gl->A[i][i] - tau);
    gl->A[i][i]-= gl->d[i];

    for (k= i+1; k < gl->n; k++) {
      gamma= (long double)gl->A[i][i] * gl->A[i][k];
      for(l= i+1; l < gl->n; l++)
        gamma+= (long double)gl->A[l][i] * gl->A[l][k];

      gamma*= beta;
      for(j= i; j < gl->n; j++)
        gl->A[j][k]+= gamma * gl->A[j][i];
    }
  }

  return;
}


void qr_ergebnis(gleichung_t *gl) {
 long double  gamma;
  int     i, j;

  //Berechnung von b
  for(i=0; i < gl->n; i++) {
    gamma= (long double)gl->A[i][i] * gl->b[i];
    for(j= i+1; j < gl->n; j++)
      gamma+= gl->A[j][i] * gl->b[j];
```

Claudia Heinich, 8966108
Sven Pohl, 8968075
Arne Brutschy, 8964813

```
    // gamma = gamma * beta
    gamma*= (long double)1/(gl->d[i] * (gl->A[i][i] + gl->d[i]) - gl->d[i] * gl->d[i]);
    for(j= i; j < gl->n; j++)
      gl->b[j]+= gamma* gl->A[j][i];
  }

  // Berechnung vom letzten x
  gl->x[gl->n-1]= gl->b[gl->n-1] / gl->d[gl->n-1];
  for(i= gl->n-2; i >= 0; i--) {
    // berechne b_i
    for(j= gl->n-1; j >= i+1; j--)
      gl->b[i]-= gl->A[i][j] * gl->x[j];

    gl->x[i]= gl->b[i] / gl->d[i];
  }

  return;
}


void zeilentausch(gleichung_t *gl, int a, int b) {
  double tmp;
  int i;

  if (a == b) return;

  for (i= 0; i < gl->n; i++) {
    tmp= gl->A[a][i];
    gl->A[a][i]= gl->A[b][i];
    gl->A[b][i]= tmp;
  }

  tmp= gl->b[a];
  gl->b[a]= gl->b[b];
  gl->b[b]= tmp;

  return;
}


void gauss(gleichung_t *gl, int sppivot) {
  int i, j, k;
  double l;

  for (i= 0; i < gl->n-1; i++) {
    // suche pivotelement
    j= i;
    // mit Spaltenpivotsuche
    if (sppivot) {
```

Claudia Heinich, 8966108
Sven Pohl, 8968075
Arne Brutschy, 8964813

```
      for (k= i + 1; k < gl-> n; k++)
        if (gl->A[k][i] > gl->A[j][i])
          j= k;
    // ohne
    } else {
      while (gl->A[j][i] == 0 && j < gl->n) j++;
    }

    if (gl->A[j][i] == 0) {
      printf("Matrix ist singulaer!\n");
      return;
    }
    zeilentausch(gl, i, j);

    // jede Zeile
    for (k= i + 1; k < gl->n; k++) {
      // faktor berechnen
      l= gl->A[k][i]/gl->A[i][i];

      // Zeile multiplizieren
      for (j= i; j < gl->n; j++)
        gl->A[k][j]= gl->A[k][j] - l * gl->A[i][j];

      gl->b[k]= gl->b[k] - l * gl->b[i];
    }
  }

  // errechne ergebnis
  for (i= gl->n-1; i >= 0; i--) {
    // errechne Wert fuer diese Zeile
    gl->x[i]= gl->b[i];
    for (j= i+1; j < gl->n; j++) {
      gl->x[i]-= gl->A[i][j];
    }
    gl->x[i]= gl->x[i]/gl->A[i][i];

    // setze Wert in alle weiteren Zeilen ein
    for (j= i+1; j >= 0; j--)
      gl->A[j][i]= gl->A[j][i] * gl->x[i];
  }
  return;
}

int main() {
  gleichung_t gl_mit;
  gleichung_t qr;


  for (gl_mit.n= 1; gl_mit.n < 21; gl_mit.n++) {
```

Claudia Heinich, 8966108
Sven Pohl, 8968075
Arne Brutschy, 8964813

```
    // erstelle Hilbert-Matrix
    hilbert(&gl_mit);
    memcpy(&qr, &gl_mit, sizeof(gl_mit));

    // berechne mittels Gauss mit Spaltenpivotsuche
    gauss(&gl_mit, 1);

    // berechne mittels QR
    qr_zerlegung(&qr);
    qr_ergebnis(&qr);

    ausgabe_ergebnis(&gl_mit, &qr);
  }

  return EXIT_SUCCESS;
}
```