

数値計算レポート 1

21D8104008F 高橋駿斗

(1) プログラムリストと説明

```
/* pivoting つき lu 分解と連立一次方程式解法 */
#include <stdio.h>
#include <math.h>

#define N 200
int lu(int n, double a[][N], int p[]){
    int i,j,k,kmax,l; double pv,t;
    for(i=0;i<n;i++) p[i]=i; /* 行の入れ替えを示す配列の初期化 */
    for(k=0;k<n;k++){ /* 枢軸選択つき反復先頭 */
        kmax=k; pv=fabs(a[k][k]); /* 枢軸候補の行番号を kmax に記憶. */
        for(i=k+1;i<n;i++) if((t=fabs(a[i][k]))>pv){ pv=t; kmax=i;}
        if(kmax!=k){ /* 第 k 行と第 kmax 行の入れ替え */
            { l=p[kmax]; p[kmax]=p[k]; p[k]=l; } /* p[kmax]と p[k]の入れ替え*/
            for(j=0;j<n;j++){ /* a[kmax][j] と a[k][j] (j=0,...,n-1)の入れ替え */
                t=a[kmax][j]; a[kmax][j]=a[k][j]; a[k][j]=t;
            }
        }
    }
    if(pv<1e-16) return 0; /* 行列 a が正則でないとみなし戻り値 0 で終了 */
    /** LU 分解本体 *****/
    for(i=k+1;i<n;i++){
        a[i][k]=a[i][k]/a[k][k];
        for(j=k+1;j<n;j++) a[i][j]=a[i][j]-a[i][k]*a[k][j];
    }
}
return 1; /* LU 分解できたので戻り値 1 */
}

void solve(int n,double a[][N],double b[],double x[],int p[]){
    int i,j,r; double s;
    r=lu(n,a,p); /* 行列 a が LU 分解できれば r == 1 となる*/
```

```

if(r==0) { printf("Can't solve¥n"); return; }

for(i=0;i<n;i++){ /* 前進消去 Ly=Pb */

    s=b[p[i]];

    for(j=0;j<i;j++) s=s-a[i][j]*x[j];

    x[i]=s; /* y_i の値を x[i] に保存 */

}

for(i=n-1;i>=0;i--){ /* 後退代入 Ux=y */

    s=x[i]; /* s=y_i */

    for(j=i+1;j<n;j++) s=s-a[i][j]*x[j];

    x[i]=s/a[i][i];

}

}

```

```

double zansa(int n,double a[][N],double b[],double x[]){

    double ans;//戻り値

    double ax[N]; //ax の行列

    int i,j;

    double cnt=0;

    double r[N];

    for(i=0;i<n;i++){

        double z=0;

        for(j=0;j<n;j++){

            z+=a[i][j]*x[j];

        }

        ax[i]=z;

    } //a*x を計算

    for(i=0;i<n;i++){

        r[i]=b[i]-ax[i];

    } //r の値を計算

    for(i=0;i<n;i++){

        cnt+=pow(r[i],2);

    } //r の二乗の和を計算

    ans = sqrt(cnt);

    //r の二乗の平方根を計算

```

```

    return ans;
}

double a[N][N];
double b[N],x[N];
int p[N];

int main(){
    int i,j,n;
    double r_norm;
    scanf("%d", &n);
    for(i=0;i<n;i++)for(j=0;j<n;j++)scanf("%lg",&a[i][j]);
    for(i=0;i<n;i++)scanf("%lg",&b[i]);

    solve(n,a,b,x,p);
    {for(i=0;i<n;i++) printf("%23.17g", x[i]);} printf("✎n");

    r_norm=zansa(n,a,b,x);
    printf("%10.4g✎n",r_norm);

    return 0;
}

```

(2) 表

	S1	S2	S3	S4	S5
d/s	3.845	7.127	6.201	3.05	4.437
D10	19.33	17.13	26.54	21.18	30.28
D25	595.6	115.5	2718	364.6	279
D50	802.3	2803	1364	4192	1.17E+04
D100	6139	7991	7075	4.77E+04	6771
D200	3.77E+04	4.85E+04	9.95E+04	1.97E+05	1.01E+05

(3) 実行時の解

D5-s1

0.19893110927370544	1.0903644054594916	-1.4778372937412372
0.62844064181375858	1.2114893769996766	

D10-s1

0.30548337726555297	0.24577765948644056	1.1293078986137015
-0.62833953120805308	0.51224185267346078	-3.634889122899573
3.5770608543662363	-1.7142489564901504	2.4994369261355591
0.6435704430883098		

D25-s1

5.7211279938387314	-17.025779112574039	6.9661503052963143
21.523352486625519	-83.857013098642028	22.913207820334108
-37.989503251301443	-9.3909759062770757	99.763470751855024
28.110815834529298	85.73218200679743	-118.19757183141543
-95.117023983641488	184.8897374549841	-0.60891634674852368
230.47198442378664	27.938790415755726	-122.73794760007705
122.48626502590847	14.020653010576238	-328.96862364638207
-141.10071948527522	-338.41171260220091	239.81109944334045
-26.558959328262745		

D50-s1

2.3743990067762271	-6.1548587330225795	-7.2283793794541795
8.1120693175988947	3.8834667163496119	1.3474897035301356
13.629332109348431	-35.467012343243709	28.490442926066237
-71.009719065600365	37.552257893650513	31.33687505335719
0.59436980548184304	29.708391603470105	-22.447614231885282

42.697380839107126	73.397480822123654	-70.978436456080402
16.905124506553907	-28.768891635019703	-60.922316600019492
-63.633653480198568	-53.453522938810238	18.298133291444763
-5.6656808859368413	75.968826499363431	-56.258319831763053
58.565293970959438	91.043015981318916	56.871796318605739
-2.7290119217277313	130.54072673907453	-91.090376062644197
97.955988660135972	-15.202844480720401	-208.82935269947527
3.0599863683498856	139.68553747227489	-293.69786953255823
22.577985595654944	67.701954809866564	-41.284225910289074
47.989986717075318	41.101775324746455	122.81886677972479
49.740641148842705	-148.18840900975744	155.45192154084165
34.491106494892293	-165.33832318741838	

D100-s1

8.406921593385368	3.236394554327072	-10.207039833571013
13.494018315087397	-72.071913477213641	78.546665635608463
6.4996490391872266	-11.912762782316792	69.867507104519802
25.126714289276261	-92.167048164783964	17.721730644078001
35.804795626455117	-114.3164894621145	19.08367587010974
29.692228073980303	88.77474168186346	42.372110413797706
-90.718630526983404	103.82951816354644	-155.0479818331288
-25.919260000073596	-126.12048545862889	-68.790788952449915
-326.63652369701936	156.82690532338219	-339.85177595968418
-353.39173216313191	-321.46732849510789	-422.87237182658833
-615.17638738880498	-6.6721095718794334	66.94155640637463
-358.57910833410494	82.359248402844912	170.86295903401776
532.89502622994212	38.374169586693249	-1.5838146888544387
29.819101142205515	292.34561185486922	125.74731698559286
213.11943231502192	284.70043403309541	29.452703657610506

-114.17347323745778	220.86284752181024	-279.48254356338134
507.68809822838682	60.849941870883569	212.9488546522239
-340.9935961031	172.28374685210261	-263.25758500544538

-626.69404663822979	148.66529737634011	-45.401498087986297
429.20517829218471	-435.99033767180487	-185.12561270481905
369.65162462053104	402.75220483848591	-69.226322195718041
79.003395470356125	766.46978957134309	-358.26507850160601
86.047731711998878	-7.4795767054635869	-341.79349462116846

-55.752026790202031	994.34701539087519	332.0336373762504
204.01431247525545	571.26222327760968	394.700025578804

-392.94219329596649	352.9394826873272	-117.13896242876162
---------------------	-------------------	---------------------

-426.43745059116509	193.26153213585482	136.5591147890903
462.67089723683915	688.91544231227203	-90.89640714160663
422.17535895943718	-660.51706911898953	1129.2659014276205

-197.28767754052046	-813.03864480999118	-1023.6884928522509
622.45095172053402	527.01688054778947	-1036.0008627410352
127.58685070060555	612.60440018142901	-33.335798839480518
256.44956937816966	-380.64696056018556	-152.73282477317613
20.614744260981436		

D200-s1

-6.0927858198926357	-12.416213831539817	-1.7999966967447967
-51.462923509575845	51.35238319151582	49.937447405056503
-15.375359241320748	46.197319360787809	43.899184370302528
-129.34339086928193	-84.795518089030978	32.930716876865233
-4.5805586157419347	7.6842054336754115	99.886235415905105
-48.522584747152095	5.2658143789152323	337.92473676096967

-398.23007700910767	260.61042062276096	32.818345863661378
-155.31264374098896	-126.97236589049744	405.37375247938672
-267.1248455670879	213.13651726465667	218.72737380732048
-162.28170732025146	-163.50178997724313	519.0780810208679
-611.7760820431912	235.81158667763452	-216.50515145024062
301.57790495103615	-236.58527137326536	481.2170650789821
252.92762030430231	77.936226109650306	-385.61824791625617
-794.12014235407776	762.31254237808344	391.85852090209738
50.984578899851464	169.23470640279106	270.59090696861665
838.55672685943125	81.854752495112365	96.791275486295589
195.71510443429389	1450.6062141140328	317.97085298746799
-181.12847045484327	-190.28791227750995	-159.98752842072705
-935.7192121868934	-489.19865313158448	-41.227437342495676
140.16973927956366	404.73665554932421	792.27177762720294
312.98427957477435	625.37587566137688	-574.87139050737221
13.502644033986858	1306.9584500860485	-635.51821122883348
-600.70523875134177	-172.14654279129522	-363.29220568271455
683.0033529405913	470.17972838779281	-1233.8959054624945
753.95177047136121	944.13435722253507	807.66883138726575
-198.3015245955477	-635.89104651827836	653.41305691044784
632.47132982450432	183.02734103781435	395.44259313421651
142.32985388596566	-174.12000112121527	-1084.1364194244288
-426.81692315664941	-180.09113199403043	-222.09518143944783
27.245525628233246	27.541970435441225	325.23008063356298
1003.0012866352913	1461.2974258346494	449.80012328244862
419.97225892947586	80.921870737144417	704.59375112190264
-1436.56115609164	333.97597338469245	41.040054773893473

-459.30469775162828	452.63756869656874	-57.627218853218999
812.15218144051062	738.51690870974119	1764.6717282788607
-847.62499537078099	-662.30546740914292	-1133.2587522829742
888.5571893227351	-1078.8465052325796	442.34212926585269
-1134.4835330767335	-444.66497922743207	-2583.6748745844834
1788.2711386042652	1728.9232220118968	662.90705710263751
-1861.1523225417131	-552.36398074019837	-543.26822031834934
-383.72531423205402	-583.82957159751095	143.66091315569983
-1857.9477047028724	-988.29340217902904	-1710.4406908475014
-2596.5217031412822	1374.6009554498453	-843.45238761669896
1451.5792731361064	518.11177217934357	1392.4625199798706
-503.63465376641767	246.8794823289914	-1235.7884477925131
-1606.6455863743956	-1069.5441130800677	-2577.3893962999923
-1029.4147058511394	-511.45671084420417	-1856.8538367024923
-906.54450838916262	-390.53867407529185	-524.68819800626864
-1722.8188180945351	-433.85930836125573	-211.51132340422643
892.0038384417453	1031.5541199134395	-2393.4858934913809
-1780.5168169095543	2095.9473296005776	-1508.9152704341955
4001.9163741378029	-1092.6784402785056	109.45008360211729
-1040.6565119856116	-1907.2198253923111	1675.0112454473224
1353.8256713756834	-2254.2077325850341	918.93808546950413
-695.98600903744034	943.32779445507208	2243.4515542224908
1476.8606226889931	-877.43933239266642	1679.7117964295308
2917.2157027335134	-1290.5762763059095	-162.22014282420034
1979.7410768876423	793.0467441281022	-203.73640190747176

-4094.3905305072053	3187.5931689161839	-0.9787724167076608
-847.62692057069967	-1542.2830971568346	-2340.8811175788501
-867.99658546778312	-1133.5271689645092	-99.531439996112482
2812.7034990520192	-420.11894443833091	3338.5987647314964
-2935.041782119145	1922.1966191130061	-3389.897858341777
-521.85323748089093	-938.50766759261296	979.68904618652903
1456.3068553617645	-881.34058096430476	-1232.4015927973387
916.38997394657633	1866.0037486013739	657.33737844437064
547.14916227533763	-2007.0656265967043	

(4) 考察

計算する行列の大きさが大きくなるほど残差が大きくなってしまっている。

D25 あたりから残差が無視できないほど大きくなっているので反復法も考慮すべきである。

(5)

lu 分解のプログラムを作ってみて効率的な計算方法がわかった。

現在では行列を簡単にかつ高速に計算するライブラリはたくさんあるがその仕組みを理解する機会を得られたのでためになった。実際のプログラムでは計算量が重視されるので参考にしたいと思った。