

```

misha@m623:~/teaching/094/math575a/hw6/legendre$ head -13 ../../matrix.c | tail -6
void create_matrix(double ***A, int m, int n) { int i;
    *A = (double **)malloc(m * sizeof(double *));
    for (i = 0; i < m; i++) (*A)[i] = (double *)malloc(n * sizeof(double)); }

void kill_matrix(double ***A, int m)
    { int i; for (i = 0; i < m; i++) free((*A)[i]); free(*A); }
misha@m623:~/teaching/094/math575a/hw6/legendre$ head -137 ../../matrix.c | tail -50
void house(double **A, double **R, double **W, int m, int n)
{ /* algorithm 10.1 */
    int i, j, k;
    double *x, tmp; x = (double *)malloc(m * sizeof(double));
    double **r; create_matrix(&r, m, n);

    for (i = 0; i < m; i++) for (j = 0; j < n; j++) r[i][j] = A[i][j];

    for (k = 0; k < n; k++)
    {
        for (i = k; i < m; i++) x[i] = r[i][k]; /* x = A_k:m,k */

        for (tmp = 0., i = k; i < m; i++) tmp += (x[i] * x[i]);
        tmp = sqrt(tmp); if (x[k] < 0) tmp = -tmp;
        x[k] += tmp; /* x = sign(x_1) |x| e_1 + x */

        for (tmp = 0., i = k; i < m; i++) tmp += (x[i] * x[i]); tmp = sqrt(tmp);
        for (i = k; i < m; i++) W[i][k] = x[i] / tmp; /* v_k = x / |x| */

        for (j = k; j < n; j++) /* A_k:m,k:n = (I - 2 W_k W_k^*) A_k:m,k:n */
        {
            for (tmp = 0., i = k; i < m; i++) tmp += W[i][k] * r[i][j];
            for (i = k; i < m; i++) r[i][j] -= 2. * tmp * W[i][k];
        }
    }

    free(x);
    for (i = 0; i < m; i++) for (j = i + 1; j < n; j++) W[i][j] = 0.;
    for (i = 0; i < n; i++) for (j = 0; j < n; j++)
        if (i > j) R[i][j] = 0.; else R[i][j] = r[i][j];
    kill_matrix(&r, m);
}

void form_Q(double **W, double **Q, int m, int n)
{ /* algorithm 10.3 */
    int i, j, k;
    double tmp;

    for (j = 0; j < n; j++)
        { for (i = 0; i < m; i++) Q[i][j] = 0.; Q[j][j] = 1.; }

    for (k = n - 1; k >= 0; k--)
    {
        for (j = k; j < n; j++) /* Q_k:m,k:n = (I - 2 W_k W_k^*) Q_k:m,k:n */
        {
            for (tmp = 0., i = k; i < m; i++) tmp += W[i][k] * Q[i][j];
            for (i = k; i < m; i++) Q[i][j] -= 2. * tmp * W[i][k];
        }
    }
}
misha@m623:~/teaching/094/math575a/hw6/legendre$

```

```

misha@m623:~/teaching/094/math575a/hw6/legendre$ cat legendre.c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include ".././matrix.c"

#define K 4

int main(int argc, char **argv)
{
    int i, j, N, scheme;
    double x, **A, **Q, **R, **W, **Qex, tmp, norm;

    N = atoi(argv[1]); scheme = atoi(argv[2]);

    create_matrix(&A, 2 * N + 1, K); create_matrix(&Qex, 2 * N + 1, K);
    for (i = 0; i <= 2 * N; i++)
    {
        x = ((double)i) / ((double)N); x -= 1.;
        A[i][0] = 1.; A[i][1] = x; A[i][2] = x * x; A[i][3] = x * x * x;
        for (A[i][0] = 1., j = 1; j < K; j++) A[i][j] = x * A[i][j - 1];

/* Qex --- exact Legendre polynomials */
        Qex[i][0] = 1.; Qex[i][1] = x;
        Qex[i][2] = (3. * x * x - 1.) / 2.;
        Qex[i][3] = (5. * x * x - 3.) * x / 2.;
    }

/* choosing the integration scheme */
    switch (scheme)
    {
        case 0: break; /* no correction */
        case 1: /* trapezoid */
            for (tmp = sqrt(0.5), j = 0; j < K; j++)
                { A[0][j] *= tmp; A[2 * N][j] *= tmp; }
            break;
        case 2: /* Simpson */
            for (j = 0; j < K; j++) { A[0][j] *= 0.5; A[2 * N][j] *= 0.5; }
            for (tmp = sqrt(0.5), j = 0; j < K; j++)
                for (i = 2; i < 2 * N; i += 2) A[i][j] *= tmp;
            break;
    }
    for (norm = 0., i = 0; i <= 2 * N; i++) norm += A[i][0] * A[i][0];
    norm = sqrt(norm);

    create_matrix(&Q, 2 * N + 1, K); create_matrix(&R, K, K);
    create_matrix(&W, 2 * N + 1, K);
/* QR factorization */
    house(A, R, W, 2 * N + 1, K); form_Q(W, Q, 2 * N + 1, K);
    for (j = 0; j < K; j++) if (R[j][j] < 0)
        { for (i = 0; i <= 2 * N; i++) Q[i][j] = -Q[i][j];
          for (i = 0; i < K; i++) R[j][i] = -R[j][i]; }

/* rescaling calculated Legendre polynomials */
    switch (scheme)
    {
        case 0: break; /* no correction */

```

```

case 1: /* trapezoid */
    for (tmp = sqrt(2.), j = 0; j < K; j++)
        { Q[0][j] *= tmp; Q[2 * N][j] *= tmp; }
    break;
case 2: /* Simpson */
    for (j = 0; j < K; j++) { Q[0][j] *= 2.; Q[2 * N][j] *= 2.; }
    for (tmp = sqrt(2.), j = 0; j < K; j++)
        for (i = 2; i < 2 * N; i += 2) Q[i][j] *= tmp;
    break;
}
for (j = 0; j < K; j++)
{
    tmp = norm / sqrt((double)(2 * j + 1));
    for (i = 0; i <= 2 * N; i++) Q[i][j] *= tmp;
}

/* printing errors */
printf("%8d ", N);
for (j = 0; j < K; j++)
{
    for (tmp = 0., i = 0; i <= 2 * N; i++)
        tmp += (Q[i][j] - Qex[i][j]) * (Q[i][j] - Qex[i][j]);
    tmp /= (double)N; tmp = sqrt(tmp); printf("%e ", tmp);
}
printf("\n");

kill_matrix(&A, 2 * N + 1); kill_matrix(&Qex, 2 * N + 1);
kill_matrix(&Q, 2 * N + 1); kill_matrix(&R, K); kill_matrix(&W, 2 * N + 1);
return 0;
}

```

```

misha@m623:~/teaching/094/math575a/hw6/legendre$ cat run
#!/bin/bash

```

```

cc legendre.c -lgsl -lgslcblas -lm

```

```

for scheme in 'seq 0 2'
do
    case $scheme in
        0 )
            echo "no correction:" ;;
        1 )
            echo "trapezoid:" ;;
        2 )
            echo "Simpson:" ;;
    esac

```

```

N=2
for i in 'seq 1 16'
do
    ./a.out $N $scheme
    N='expr $N \* 2'
done
echo ""

```

```

done
misha@m623:~/teaching/094/math575a/hw6/legendre$

```

```
misha@m623:~/teaching/094/math575a/hw6/legendre$ ./run
```

```
no correction:
```

```
 2 4.440892e-16 2.051631e-01 4.873433e-01 7.391343e-01
 4 6.377746e-16 1.022204e-01 2.412662e-01 3.739626e-01
 8 3.510833e-16 5.105314e-02 1.197159e-01 1.858620e-01
16 3.188873e-16 2.551845e-02 5.958852e-02 9.232583e-02
32 1.177569e-16 1.275814e-02 2.972194e-02 4.596876e-02
64 1.665335e-16 6.378927e-03 1.484228e-02 2.293037e-02
128 6.069053e-16 3.189446e-03 7.416388e-03 1.145099e-02
256 4.163336e-17 1.594721e-03 3.706997e-03 5.721858e-03
512 1.573156e-16 7.973600e-04 1.853198e-03 2.860009e-03
1024 3.371617e-16 3.986800e-04 9.265239e-04 1.429773e-03
2048 7.241925e-16 1.993400e-04 4.632431e-04 7.148287e-04
4096 4.355642e-16 9.966999e-05 2.316168e-04 3.573998e-04
8192 1.570092e-16 4.983500e-05 1.158072e-04 1.786963e-04
16384 2.602085e-18 2.491750e-05 5.790333e-05 8.934723e-05
32768 6.133174e-18 1.245875e-05 2.895159e-05 4.467339e-05
65536 5.368374e-16 6.229375e-06 1.447578e-05 2.233664e-05
```

```
trapezoid:
```

```
 2 4.902612e-16 6.394144e-02 2.584883e-01 5.635652e-01
 4 4.041273e-16 1.478325e-02 6.650978e-02 1.767163e-01
 8 3.845925e-16 3.466726e-03 1.560473e-02 4.429819e-02
16 9.945640e-16 8.334226e-04 3.672893e-03 1.039670e-02
32 3.682193e-16 2.039314e-04 8.837442e-04 2.464164e-03
64 3.152427e-16 5.041401e-05 2.162881e-04 5.962523e-04
128 1.679728e-16 1.253144e-05 5.347130e-05 1.464236e-04
256 1.962616e-17 3.123793e-06 1.329155e-05 3.626622e-05
512 8.398642e-17 7.798112e-07 3.313280e-06 9.023510e-06
1024 5.495762e-16 1.948105e-07 8.271145e-07 2.250461e-06
2048 1.816082e-16 4.868481e-08 2.066277e-07 5.619359e-07
4096 5.467324e-16 1.216898e-08 5.163803e-08 1.403989e-07
8192 6.201873e-16 3.041965e-09 1.290714e-08 3.508909e-08
16384 3.140233e-16 7.604522e-10 3.226490e-09 8.770941e-09
32768 4.540206e-17 1.901098e-10 8.065836e-10 2.192567e-09
65536 9.428765e-16 4.752685e-11 2.016355e-10 5.481197e-10
```

```
Simpson:
```

```
 2 7.850462e-17 2.237726e-16 4.679670e-02 3.624944e-01
 4 5.087681e-16 1.798767e-16 2.461114e-03 3.273475e-02
 8 9.330168e-16 2.575816e-16 1.345908e-04 1.829267e-03
16 5.125396e-16 3.274908e-16 7.815468e-06 1.030385e-04
32 1.177569e-16 5.539312e-16 4.702367e-07 6.070502e-06
64 5.833625e-16 1.962870e-16 2.882834e-08 3.678840e-07
128 5.151632e-16 2.423454e-16 1.784366e-09 2.263463e-08
256 5.892751e-16 2.810325e-16 1.109814e-10 1.403514e-09
512 2.693239e-16 2.356006e-16 6.918993e-12 8.737216e-11
1024 2.989568e-16 3.533255e-16 4.309129e-13 5.449066e-12
2048 1.226635e-16 6.209893e-16 2.645158e-14 3.405635e-13
4096 1.736977e-16 1.913663e-15 6.390192e-15 2.338473e-14
8192 6.148390e-16 8.921969e-16 4.598077e-15 5.007531e-15
16384 6.206329e-16 2.814532e-16 1.821979e-14 6.248994e-15
32768 5.055715e-16 6.554841e-16 1.148364e-14 7.376122e-15
65536 5.878093e-16 3.784174e-16 2.231417e-14 1.440652e-14
```

```
misha@m623:~/teaching/094/math575a/hw6/legendre$
```