Numerical Computations - Ex 5, Nov 12, 2020

17. Find Gaussian integration rules of arbitrary order.
    Use that the monic Legendre polynomials are

$$\tilde{L}_n(x) = \det(xI - A_n)$$

with

$$A_n = \begin{pmatrix} 0 & \gamma_1 & 0 & \dots & 0 \\ \gamma_1 & 0 & \gamma_2 & \ddots & \vdots \\ 0 & \gamma_2 & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \gamma_{n-1} \\ 0 & \cdots & 0 & \gamma_{n-1} & 0 \end{pmatrix}$$

and

$$\gamma_j = \frac{1}{\sqrt{4 - j^{-2}}}.$$

Verify the formulas by plotting Legendre polynomials $L_n(x)$, and $\tilde{L}_n(x)/\tilde{L}_n(1)$.

Find roots of $\tilde{L}_n$ by computing the eigenvalues of A_n.

Find integration weights by solving a linear system of equations. The integration rule with $n + 1$ points must be exact (at least) for polynomials up to order n. To set up the equation, use either monomials x^i, or Legendre polynomials. Test your integration rules as in Ex 15. How far can you go with n ?

You may use libraries for finding eigenvalues (scipy.linalg.eigh), solving linear systems, computing determinants, and evaluating Legendre polynomials.

nice reading: https://gubner.ece.wisc.edu/gaussquad.pdf, Example 15

18. Find quadrature rules for the circle.
    Let $K = \{(x, y) : x^2 + y^2 \leq 1\}$. Find integration points (x_i, y_i) and integration weights ω_i such that

$$\int_K f(x, y) d(x, y) \approx \sum_{i=0}^{n-1} \omega_i f(x_i, y_i)$$

Hint: Use polar coordinates $x = r\cos\varphi, y = r\sin\varphi$ to transform the integral to a rectangular integration domain:

$$\int_K f(x, y) d(x, y) = \int_0^1 \int_0^{2\pi} r f(r\cos\varphi, r\sin\varphi)\, d\varphi\, dr$$

Compare different rules for the r-integral and φ-integral.
Test for functions $f(x, y) = 1$, $f(x, y) = x^3 y^4$, $f(x, y) = \exp(-(x-1)^2 - y^2)$.

19. Calculate the LU decomposition of the matrix

$$A = \begin{pmatrix} 10^{-6} & 1 \\ 1 & 0 \end{pmatrix}$$

by hand. Find a normalized lower left triangular matrix L and an upper right triangular matrix U such that

$$A = LU.$$

Furthermore, calculate A^{-1}, L^{-1}, and U^{-1}, and the condition numbers

$$\mathrm{cond}_\infty(M) = \|M\|_\infty \|M^{-1}\|_\infty$$

for all of these matrices. The maximum absolute row sum norm (Zeilenbetragssummennorm) is defined as

$$\|M\|_\infty = \max_{i=1,\dots n} \sum_{j=1}^{n} |M_{i,j}|.$$

Then, swap the rows of A and repeat the calculations.

20. Measure timings for

    (a) inner products of vectos $x \cdot y$

    (b) matrix vector products $y = Ax$

    (c) matrix matrix products $A = BC$

    where x and y are vectors in $\mathbb{R}^n$, and A, B, C are $n \times n$ matrices. Plot the number of floating point operations per second you have measured.

    Compare different programming languages and libraries:

    - loops written in Python
    - numpy matrices
    - hand-written C++ loops
    - C++ library Eigen: http://eigen.tuxfamily.org
    - C++ library Blaze: https://bitbucket.org/blaze-lib/blaze/src/master/
    - C++ library ngbla: https://github.com/NGSolve/ngbla

    Choose at least three of the list. Repeat the operations for small matrices and vectors to obtain accurate results. Go up to matrix / vectors sizes such that the time is in the range of seconds (if you have enough memory).