Numerical Approximation Extra-credit Project (Optional)

due 2020 JUN 16, 9:50 a.m.

Write a program to implement the algorithm for poised-lattice generation and validate your implementation for $\mathrm{D} = 2, 3$ and $n = 2, 3, 4, 5$.

Your submission should consist of the following.

(i) A `C++` package that implement the PLG algorithm in the notes. In your package, you should have a number of classes, each of which takes on a single responsibility. In each class, you should clearly state the input, output, preconditions (if applicable), postconditions (if applicable), and side-effects (if applicable). You should comment your code so that it is easy to follow your logic.

(ii) Two-dimensional tests and results as exemplified by Figure 1.

(iii) Three-dimensional tests and results as exemplified by Figure 2. In addition to the test ordering in the notes, you should also implement another testing ordering

$$\forall j, k \in J_T,\ j < k \ \Leftrightarrow\ \big(|j - q_i| > |k - q_i|\big) \vee \big(|j - q_i| = |k - q_i| \ \wedge \ \#L_{i,j}(K) < \#L_{i,k}(K)\big). \tag{1}$$

(iv) Tests of bi-variate and tri-variate interpolation of functions

$$\mathbf{u}(x, y) = \begin{pmatrix} \sin^2(\pi x)\sin(2\pi y) \\ -\sin(2\pi x)\sin^2(\pi y) \end{pmatrix}, \tag{2}$$

$$\mathbf{u}(x, y, z) = \frac{1}{2}\begin{pmatrix} \sin^2(\pi x)\sin(2\pi y)\sin(2\pi z) \\ \sin(2\pi x)\sin^2(\pi y)\sin(2\pi z) \\ -2\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z) \end{pmatrix} \tag{3}$$

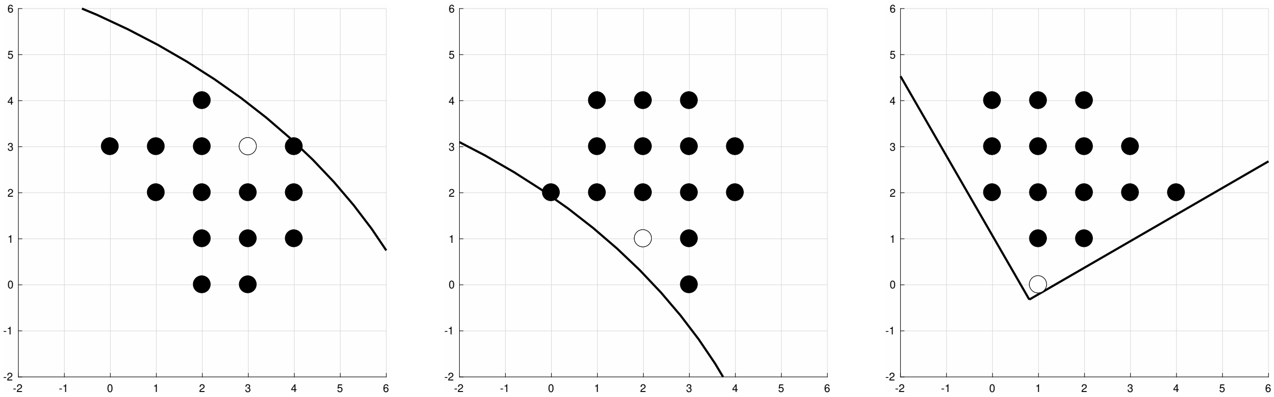
on the triangular lattices you generated in (ii) and (iii). Output error norms (L_1 and L_∞) and convergence rates at the centers of squares and cubes within $\mathbb{Z}_n^{\mathrm{D}}$, with the interpolation sites at vertices of the squares and cubes. The higher degree of the interpolating polynomial, the higher convergence rates you should get for the interpolation.

(v) A `test` option in your `Makefile` so that the command `make test` will run all tests and generate all results.

(vi) A design document typeset in LaTeX explaining the design of your `C++` package and the correspondence between math and `C++` classes. The design document should also contain all the details that make your tests unique.

This extra-credit project is purely optional. Your submission will first be graded on a scale of 70 points, then *any score lower than 50 will be set to zero*! This means that either good work or no work will be accepted as extra credit. So you need to think twice before you invest substantial time and effort in this project.

Please upload your source files, supporting documents, and program code in a single zip file to the course email `NumApproximation@163.com`; the name of your zip file should be `YourName_finalProject.zip`.

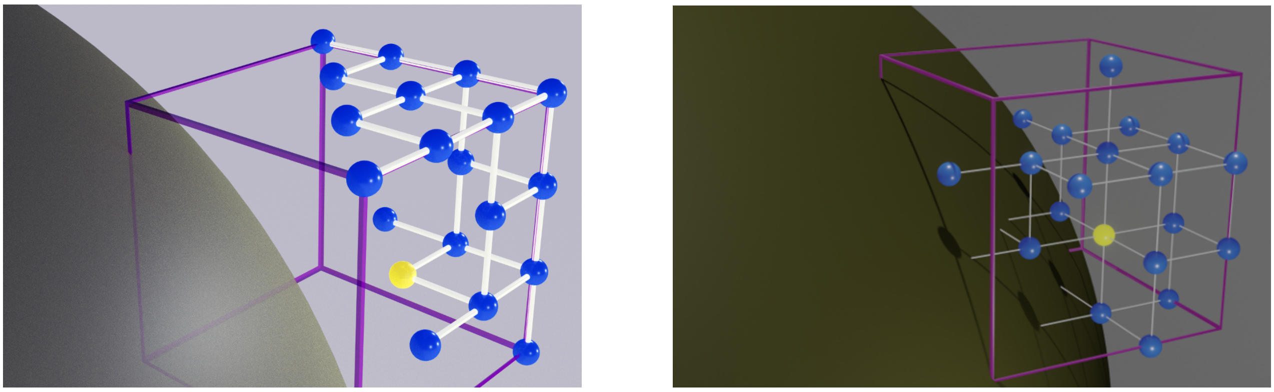


(a) Near a concave interface

(b) Near a convex interface

(c) At the corner of a rotated box

Figure 1: 2D tests of the PLG algorithm with $n = 4$. The hollow circle is the starting point $\mathbf{q}$. The irregular interfaces in the first two subplots are given by $\frac{(x-x_0)^2}{a^2} + \frac{(y-y_0)^2}{b^2} = 1$.



(a) Using the test ordering in the notes

(b) Using the test ordering (1)

Figure 2: 3D tests of the PLG algorithm with $n = 3$. The yellow circle is the starting point $\mathbf{q}$. The irregular interface is given by an ellipsoid.