GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

Scientific Computing

Polynomial Approximation

Prepared by

Abdul Halim

Project Statement

This project, led by Abdul Halim, delves into the sophisticated realm of adaptive polynomial approximation for multivariate functions using advanced algorithms and tree-based data structures. The focus lies on developing efficient, accurate, and scalable methods to approximate complex functions defined on multi-dimensional domains, leveraging Python, MATLAB, and mathematical principles. This work embodies cutting-edge advancements in scientific computing, bridging computational mathematics and software development.

Purpose

The primary goal of this project is to create a computational framework capable of approximating functions $f : [0, 1]^n \to \mathbb{R}$ with high precision while minimizing computational overhead. This is achieved through adaptive refinement techniques, where computational effort is focused on regions requiring higher accuracy. Such methodologies are pivotal in solving partial differential equations (PDEs), data interpolation, computer graphics, and physical simulations.

Key objectives include:

1. **Algorithm Development:**
    - **Threshold-Based Adaptive Refinement:** A baseline approach that uses fixed thresholds for mesh refinement.
    - **Greedy Error Minimization:** An optimization-driven method for selecting mesh subdivisions.
    - **General Adaptive Strategy:** A dynamic method handling overlapping cells for higher-dimensional function approximations.
2. **Error Estimation:**
    - Design error estimators using second derivatives and residual calculations to identify regions needing finer approximation.
    - Employ L^p-norms to quantify approximation accuracy.
3. **Triangulation and Mesh Refinement:**
    - Develop an adaptive mesh refinement strategy using quadtrees (2D) and octrees (3D).
    - Implement continuous and discontinuous approximation modes, accommodating various application needs.
4. **Visualization and Analysis:**
    - Provide visual representations of the adaptive refinement process and approximation errors.
    - Conduct convergence analysis comparing adaptive and uniform refinement.

Methods

Programming and Tools

- **Python:** Libraries such as NumPy, SciPy, and Matplotlib are utilized for efficient computation and visualization.
- **MATLAB:** Deployed for prototyping algorithms and validating numerical results.
- **Tree Data Structures:** Quadtree and octree implementations form the backbone of the adaptive algorithms, enabling hierarchical refinement.

Function Approximations

Functions tested include:

- Smooth functions (e.g., Gaussian functions).
- Oscillatory functions (e.g., $\sin(4\pi x)$).
- Discontinuous functions, demonstrating robustness across diverse scenarios.

Applications

This work has broad applicability across scientific and engineering domains:

- **Numerical Solutions of PDEs:** Adaptive techniques enhance efficiency in finite element and finite volume methods.
- **Data Interpolation:** Accurate reconstruction of data from sparse samples.
- **Computer Graphics:** Generating smooth surfaces and textures in 3D modeling.
- **Physical Simulations:** Improved accuracy in simulating complex phenomena, such as fluid dynamics and structural mechanics.

Impact and Future Directions

This project exemplifies advanced problem-solving in scientific computing, showcasing the interplay of mathematics and programming to address real-world challenges. Moving forward, potential directions include extending the framework to irregular domains, integrating machine learning for automated error estimation, and parallelizing algorithms for large-scale computations.

Abdul Halim's work not only contributes to the field of adaptive approximation but also serves as a testament to the power of computational tools like Python and MATLAB in modern mathematical research.