NumPy (Numerical Python) and the `math` module in Python both deal with mathematical operations, but they have some key differences:

1. **Functionality**:
   - **NumPy**: Provides a comprehensive set of numerical computing tools, including array operations, linear algebra, Fourier transform, random number generation, and more. It's designed to work efficiently with large arrays and matrices of numeric data.
   - **math Module**: Offers a smaller set of mathematical functions compared to NumPy. It includes basic mathematical operations (e.g., trigonometric functions, logarithms, exponentials) and constants (e.g., π, e).
2. **Data Structures**:
   - **NumPy**: Introduces powerful array objects (‘`ndarray’`) that allow for vectorized operations and efficient handling of large datasets. NumPy arrays are homogeneous and support multidimensional operations.
   - **math Module**: Operates on scalar values (single numbers) and does not provide data structures for handling arrays or matrices.
3. **Performance**:
   - **NumPy**: Optimized for numerical computations and is typically faster than using Python's built-in math functions, especially when dealing with large datasets.
   - **math Module**: Since it operates on scalar values, it's suitable for basic mathematical operations but may not perform as well as NumPy for vectorized operations.
4. **Usage**:
   - **NumPy**: Widely used in scientific computing, data analysis, machine learning, and other domains where numerical computations are prevalent.
   - **math Module**: Useful for general-purpose mathematical operations in Python scripts and applications.

In summary, NumPy provides a more extensive and efficient toolkit for numerical computing compared to the `math` module. If you're working with arrays, matrices, or large datasets, NumPy is the preferred choice. However, for simple mathematical operations on individual values, the `math` module suffices.