Data Structures in MATLAB

Lecture 2

Swapnil Singh
LB, KTU

November 27, 2024

Overview

- Numerical Arrays: Foundation of MATLAB
- Cell Arrays: Heterogeneous Data Containers
- Structures: Named Fields for Organized Data
- Tables: Database-like Organization
- Advanced Data Types: Categorical & DateTime

Numerical Arrays

- Matrices and vectors are MATLAB's primary data type
- Support for multiple numeric classes
- Efficient memory management

```
% Creating arrays
A = [1 2 3; 4 5 6];
B = zeros(3,4);
C = rand(2,2);
```

Cell Arrays

Versatile Containers for Mixed Data Types

```
% Cell array creation
data = {[1 2 3], 'text', ...
        struct('field1', 10)};

% Accessing elements
element1 = data{1};  % Content
element2 = data(2);  % Cell
```

Key Features:

- Store different types in each cell
- Flexible indexing methods
- Dynamic size adjustment

Structures

Organization:

- Named fields
- Hierarchical data
- Multiple records

```
student.name = 'John';
student.grades = [85 92 78];
student.info.age = 20;
student.info.id = 'A123';
```

Tables

Database-Style Data Organization

```
% Create a table
T = table([1;2;3], ...
          {'A';'B';'C'}, ...
          [10;20;30], ...
    'VariableNames', ...
    {'ID','Name','Score'});
```

- Self-describing data structure
- Built-in support for missing data
- SQL-like operations

Advanced Data Types

Categorical Arrays

```
categories = categorical(...
    {'Small','Medium','Large'});
```

DateTime Arrays

```
dates = datetime(...
    '2024-01-01','2024-12-31');
```

- Optimized memory usage
- Special comparison operations
- Built-in analysis functions

Performance Considerations

- Preallocate arrays for better performance
- Use appropriate data type for memory efficiency
- Consider vectorization over loops
- Use sparse matrices for large, sparse data

```
% Good practice
arr = zeros(1000, 1);
for i = 1:1000
    arr(i) = i^2;
end
```

Summary

- Arrays form the foundation of MATLAB computing
- Cell arrays provide flexibility for mixed data types
- Structures organize data with named fields
- Tables offer database-like functionality
- Advanced types support specialized applications
- Choose appropriate structure based on:
  - Data characteristics
  - Performance requirements
  - Code readability