MATLAB to iPhone Made Easy

Generating readable and portable C code from your MATLAB algorithms for your iPhone or iPad app

Bill Chou
for k=1:max
    x = fft(dat)
    y = 20*log1
for k=1:max
x = fft(dat)
y = 20*log1
Quick Demo

>> Demo
Agenda

- Using MATLAB Coder
  - Challenges of manual translation
  - Two-step workflow for generating code

- Integrating Generated Code into iOS App
Challenges with Manual Translation from MATLAB to C to iOS

- Separate functional and implementation specification
  - Leads to multiple implementations that are inconsistent
  - Hard to modify requirements during development
  - Difficult to keep reference MATLAB code and C code in sync

- Manual coding errors

- Time-consuming and expensive process
Automatic Translation from MATLAB to C to iOS

With MATLAB Coder, design engineers can:

- Maintain one design in MATLAB
- Design faster and get to C quickly
- Test more systematically and frequently
- Spend more time improving algorithms in MATLAB
Simple Demo
\[ c = a \times b \]

- Code generation readiness tool
- Autodefine input type
- Code generation report

>> Demo
Using MATLAB Coder: Two-Step Workflow

Prepare your MATLAB algorithm for code generation
- Assess code readiness
- Make implementation choices
- Use supported language features

Iterate to Optimize generated C Code
- Test and verify generated C code matches behavior of MATLAB code
- Iterate MATLAB code to optimize
Implementation Considerations

function a = \textbf{foo}(b, c)
\[
a = b \times c;
\]

\begin{itemize}
\item \textbf{Element by element multiply}
\item \textbf{Dot product}
\item \textbf{Matrix multiply}
\end{itemize}

\textbf{C}

double \textbf{foo}(double b, double c)
\{
    return b * c;
\}

void \textbf{foo}(\textbf{const double} b[15],
\textbf{const double} c[30], \textbf{double} a[18])
\{
    int i0, i1, i2;
    \textbf{for} (i0 = 0; i0 < 3; i0++)
    \{
        \textbf{for} (i1 = 0; i1 < 6; i1++)
        \{
            \textbf{a}[i0 + 3 \times i1] = 0.0;
            \textbf{for} (i2 = 0; i2 < 5; i2++)
            \{
                \textbf{a}[i0 + 3 \times i1] += \textbf{b}[i0 + 3 \times i2] \times \textbf{c}[i2 + 5 \times i1];
            \}
        \}
    \}
\}
Implementation Considerations

- Polymorphism
- Memory allocation
- Processing matrices and arrays
- Fixed-point data types

7 Lines of MATLAB
105 Lines of C
Example: Newton/Raphson Algorithm

Preparing your MATLAB code

- Pre-allocate
- Identify more efficient constructs
- Select code generation options

\[ x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \]

>> Demo
MATLAB Language Support for Code Generation

- Visualization
- Java
- Nested functions
- Cell arrays
- Graphics
- Sparse
- Variable-sized data
- Struct
- Global
- Complex
- Classes
- Functions
- Malloc
- Numeric
- System objects
- Fixed-point
- Arrays
- Persistent
## Supported MATLAB Language Features and Functions

Broad set of language features and functions/system objects supported for code generation

<table>
<thead>
<tr>
<th>Matrices and Arrays</th>
<th>Data Types</th>
<th>Programming Constructs</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Matrix operations</td>
<td>• Complex numbers</td>
<td>• Arithmetic, relational, and logical operators</td>
<td>• MATLAB functions and subfunctions</td>
</tr>
<tr>
<td>• N-dimensional arrays</td>
<td>• Integer math</td>
<td>• Program control (if, for, while, switch)</td>
<td>• Variable-length argument lists</td>
</tr>
<tr>
<td>• Subscripting</td>
<td>• Double/single-precision</td>
<td></td>
<td>• Function handles</td>
</tr>
<tr>
<td>• Frames</td>
<td>• Fixed-point arithmetic</td>
<td></td>
<td>Supported algorithms</td>
</tr>
<tr>
<td>• Persistent variables</td>
<td>• Characters</td>
<td></td>
<td>• More than 700 MATLAB operators and functions</td>
</tr>
<tr>
<td>• Global variables</td>
<td>• Structures</td>
<td></td>
<td>• More than 300 System objects for:</td>
</tr>
<tr>
<td></td>
<td>• Numeric class</td>
<td></td>
<td>• Signal processing</td>
</tr>
<tr>
<td></td>
<td>• Variable-sized data</td>
<td></td>
<td>• Communications</td>
</tr>
<tr>
<td></td>
<td>• MATLAB Class (MCOS)</td>
<td></td>
<td>• Computer vision</td>
</tr>
<tr>
<td></td>
<td>• System objects</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Agenda

- Using MATLAB Coder
  - Challenges of manual translation
  - Two-step workflow for generating code

- Integrating Generated Code into iOS App
Demo: Integrating Generated Code into iOS App
More Information

- To learn more, visit the product page:
  mathworks.com/products/matlab-coder

- To request a trial license:
  - Talk to your MathWorks account manager to request a trial license and set up a guided evaluation with an application engineer
Q&A