Optimizing and Accelerating Your MATLAB Code

Debbi Cohen
RPI Account Manager

Adam Sifounakis
Application Engineer

March 30, 2016

2015 NASA Software Award – Orion GN&C

- Orion GN&C Flight Software for Exploration Flight Test 1 (EFT-1) was selected for NASA Software of the Year award this year

- Key highlights;
  - Created NASA – Orion GN&C: MATLAB and Simulink Standards
    - Supported model interoperability and code generation
  - Generated over 60K lines of code by CDR
  - Developed more accurate control algorithms that met project schedule

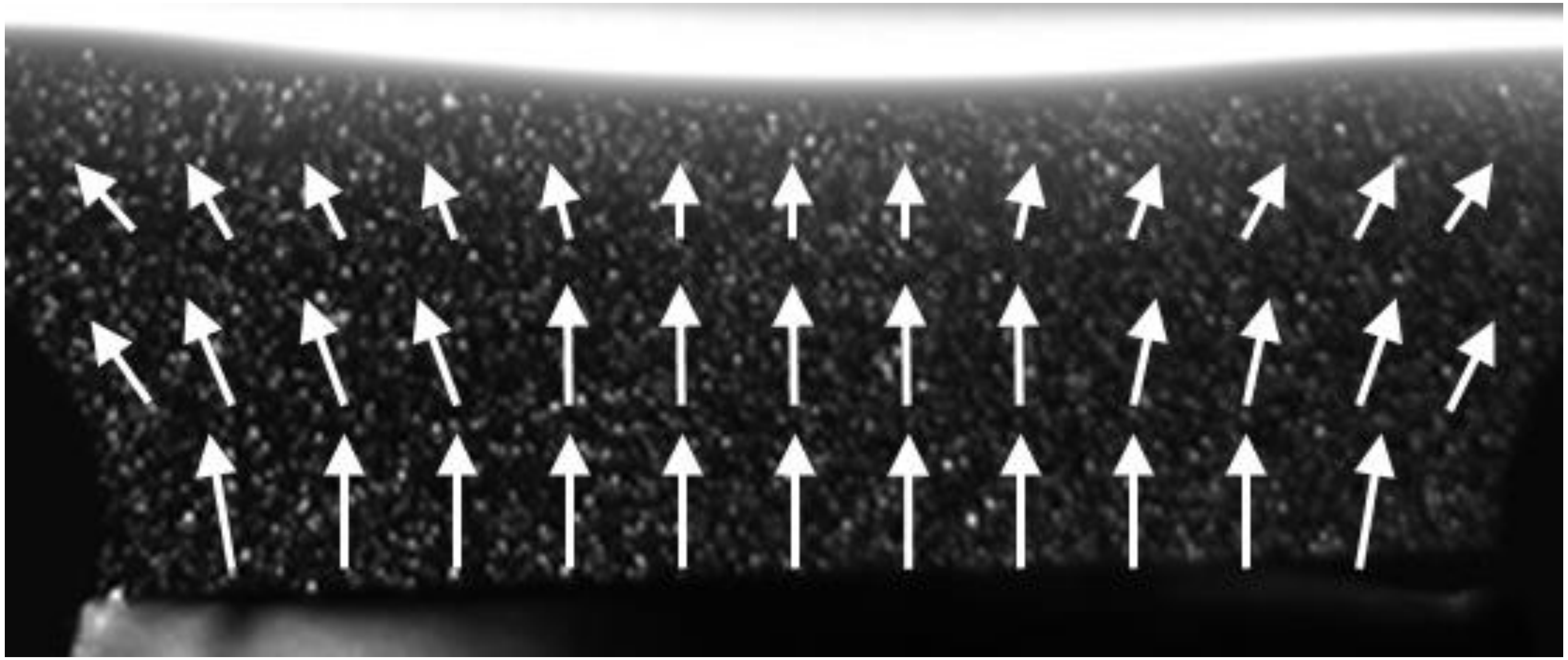
Example Projects With MathWorks

- Customers using Simulink interface to Goddard cFE software:
  - APL
  - Cornell University Space Systems Design Studio
  - NASA Ames

- Recent projects:
  - Cornell University Space Systems Design Studio – VIOLET (in progress)
  - Goddard – GEDI (in progress)
  - Goddard – NICER (in progress)

- Completed projects:
  - Ames – LADEE
    - Heavily involved with onboard flight software
  - Boeing – X40A
  - Ames – SPHERES
  - Lockheed Martin – IRIS Satellite
  - JPL – MER Rovers
  - Lockheed Martin – Mars Reconnaissance Orbiter
  - JPL – Deep Space 1

Laminar Flame Speed Calculations

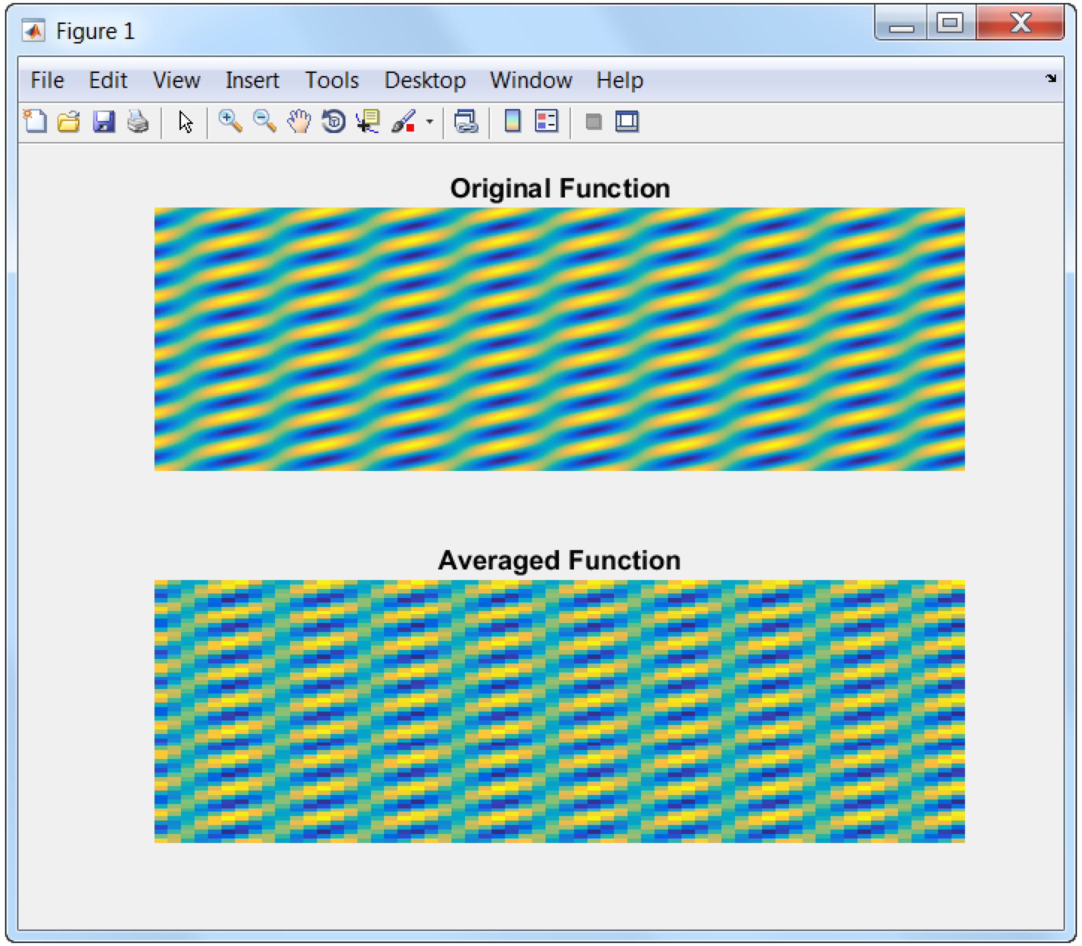


Agenda

- Optimizing `for` loops and using vector and matrix operations
- Finding and addressing bottlenecks
- Generating C code and incorporating it into your application
- Utilizing additional hardware and processing power
- Summary and resources

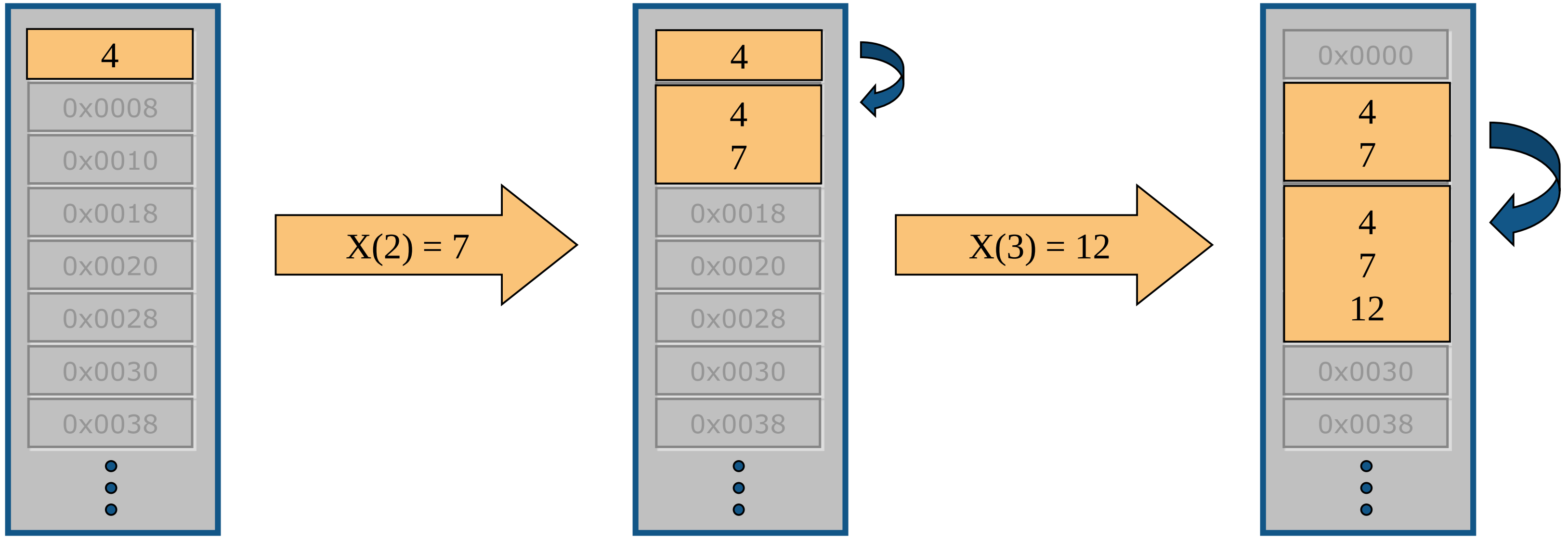
Example: Block Processing Images

- Calculate a function at grid points
- Take the mean of larger blocks
- Analyze and improve performance



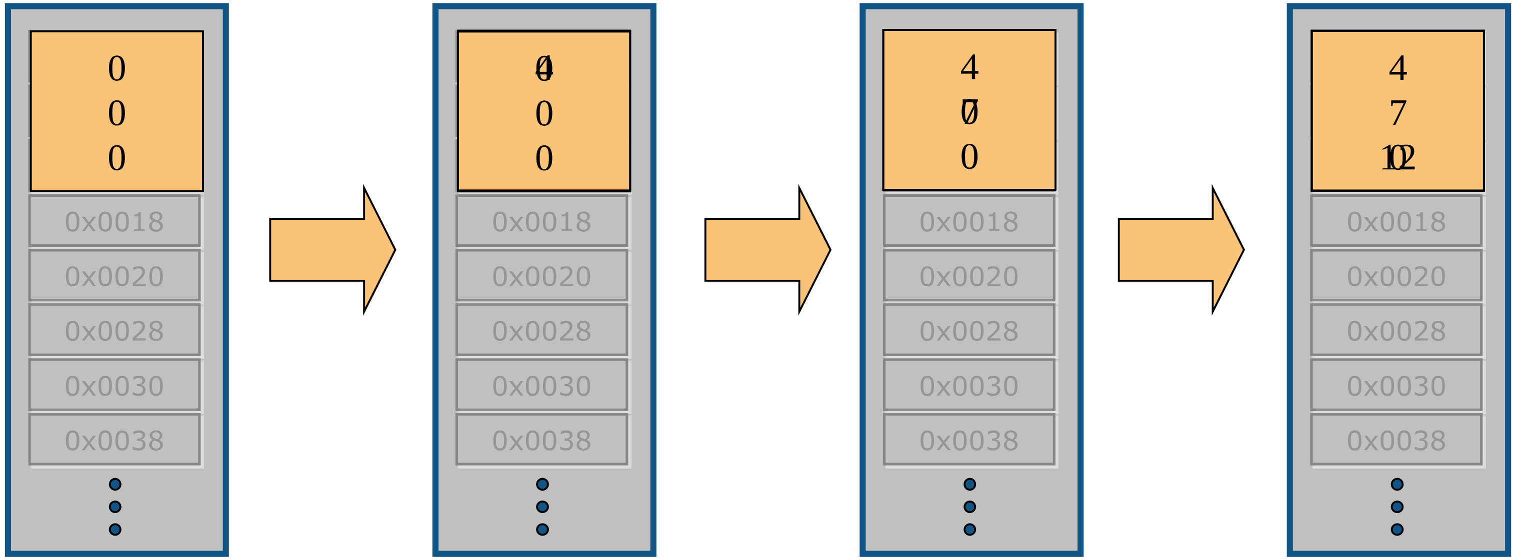
Effect of Not Preallocating Memory

```
x(1) = 4
x(2) = 7
x(3) = 12
```



Benefit of Preallocation

```
x = zeros(3,1)
x(1) = 4
x(2) = 7
x(3) = 12
```



MATLAB Underlying Technologies

- Execution Engine (>=R2015b)
  - All MATLAB code is just-in-time compiled
  - Improves “Nth run” performance

- Commercial Libraries
  - BLAS: Basic Linear Algebra Subroutines
  - LAPACK: Linear Algebra Package
  - IPP: Intel Performance Primitives
  - FFTW: Fastest Fourier Transform in the West

Other Best Practices

- Avoid “`clear all`”
  - Use “`clear`” or “`clearvars`” instead
- Use functions instead of scripts
- Keep files to less than 500 lines
- Avoid “introspection” functions
  - E.g. “`dbstack`”, “`inputname`”, “`exist`”, “`whos`”

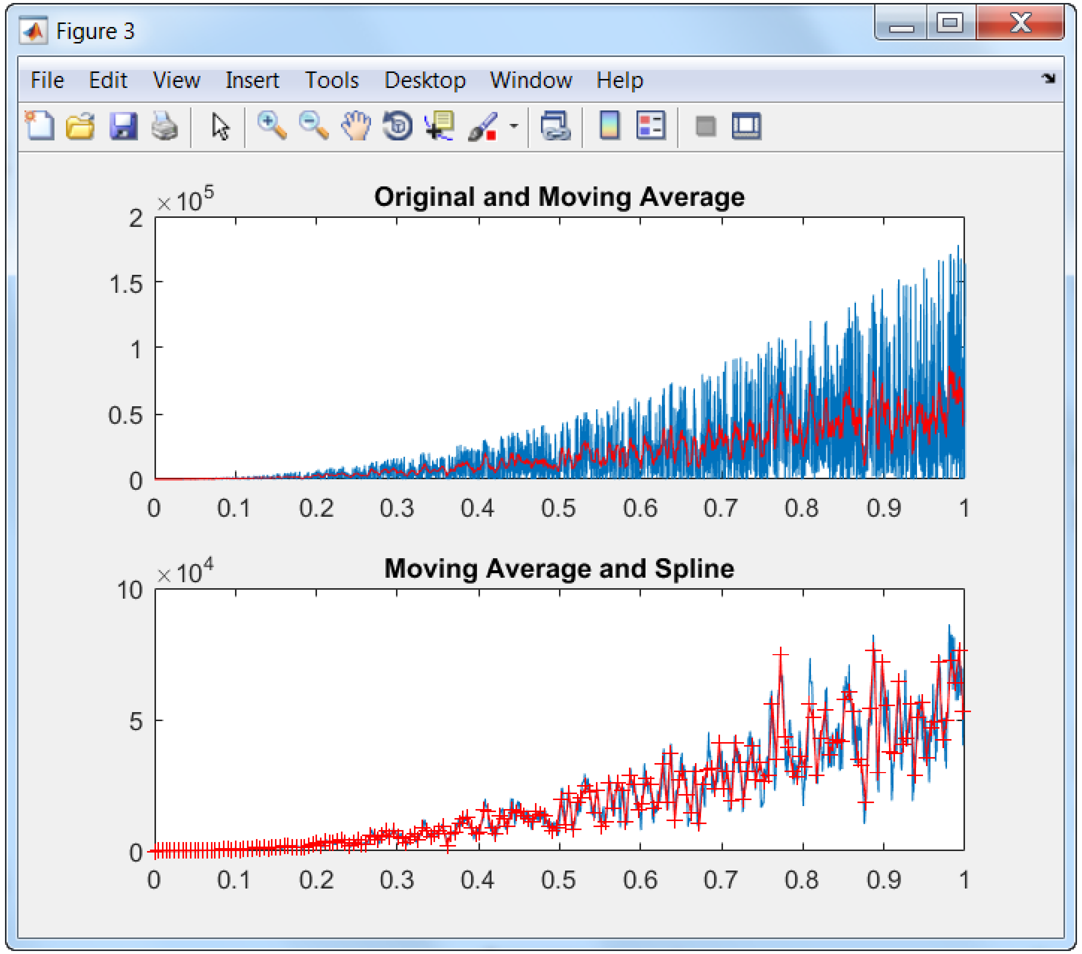
http://www.mathworks.com/help/releases/R2015b/matlab/matlab_prog/techniques-for-improving-performance.html

Agenda

- Optimizing `for` loops and using vector and matrix operations
- Finding and addressing bottlenecks
- Generating C code and incorporating it into your application
- Utilizing additional hardware and processing power
- Summary and resources

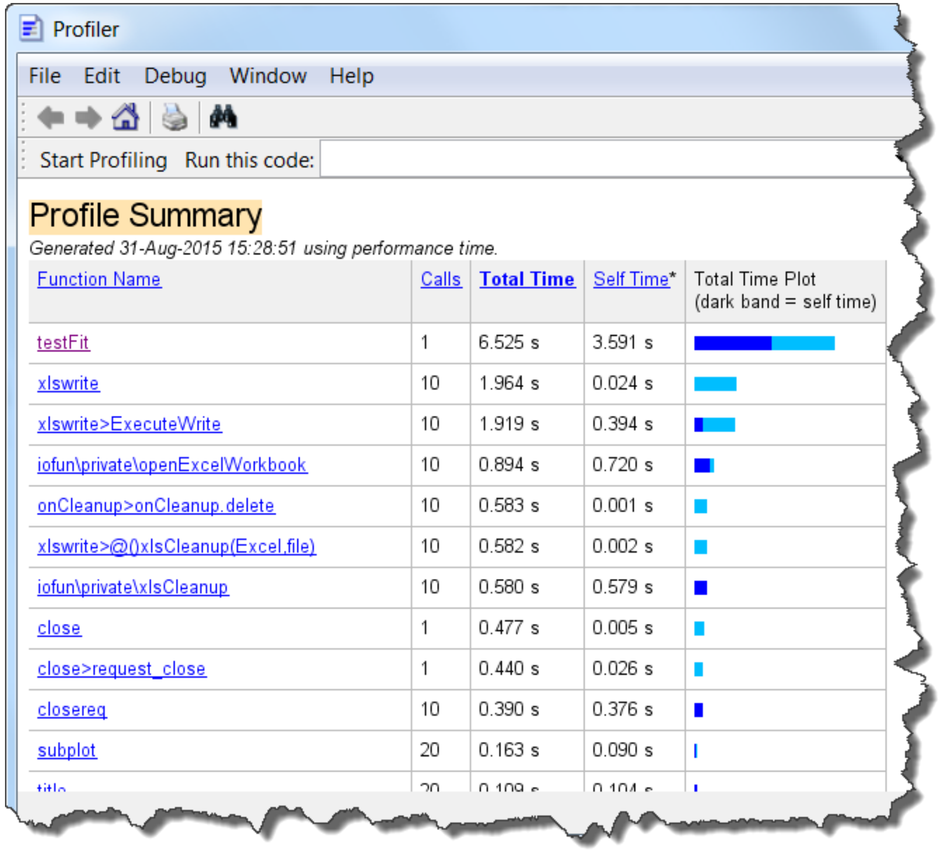
Example: Block Processing Images

- Run and time program
- Identify bottlenecks
- Improve run time



Profiler

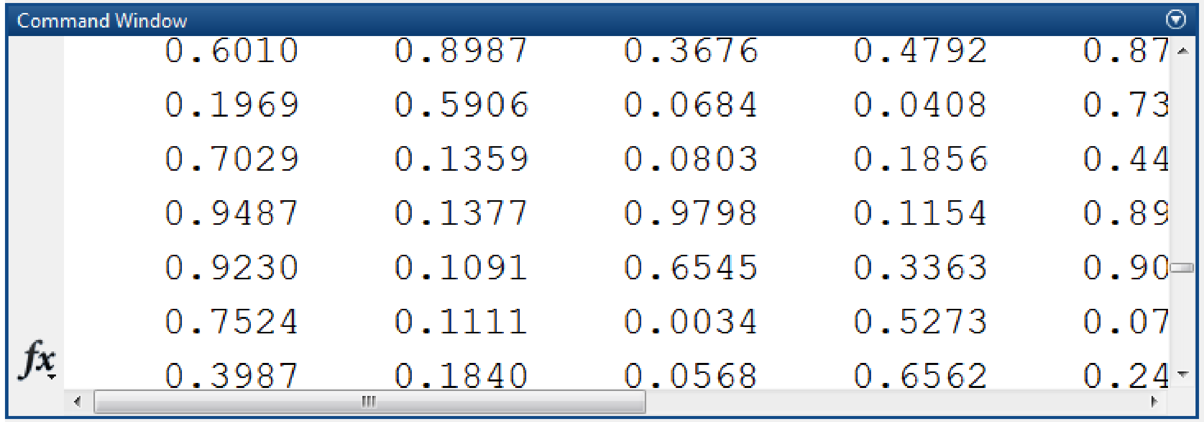
- Total number of function calls
- Time per function call
- Self time in a function call
- Code coverage



Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
testFit	1	6.525 s	3.591 s	
xlswrite	10	1.964 s	0.024 s	
xlswrite>ExecuteWrite	10	1.919 s	0.394 s	
iofun\private\openExcelWorkbook	10	0.894 s	0.720 s	
onCleanup>onCleanup.delete	10	0.583 s	0.001 s	
xlswrite>@()xlsCleanup(Excel,file)	10	0.582 s	0.002 s	
iofun\private\xlsCleanup	10	0.580 s	0.579 s	
close	1	0.477 s	0.005 s	
close>request_close	1	0.440 s	0.026 s	
closereq	10	0.390 s	0.376 s	
subplot	20	0.163 s	0.090 s	
title	20	0.109 s	0.104 s	

Best Practices

- Minimize file I/O
- Reuse existing graphics components
- Avoid printing to Command Window



Steps for Improving Performance

- First get code working
- Speed up code with core MATLAB
- Include compiled languages and additional hardware

Agenda

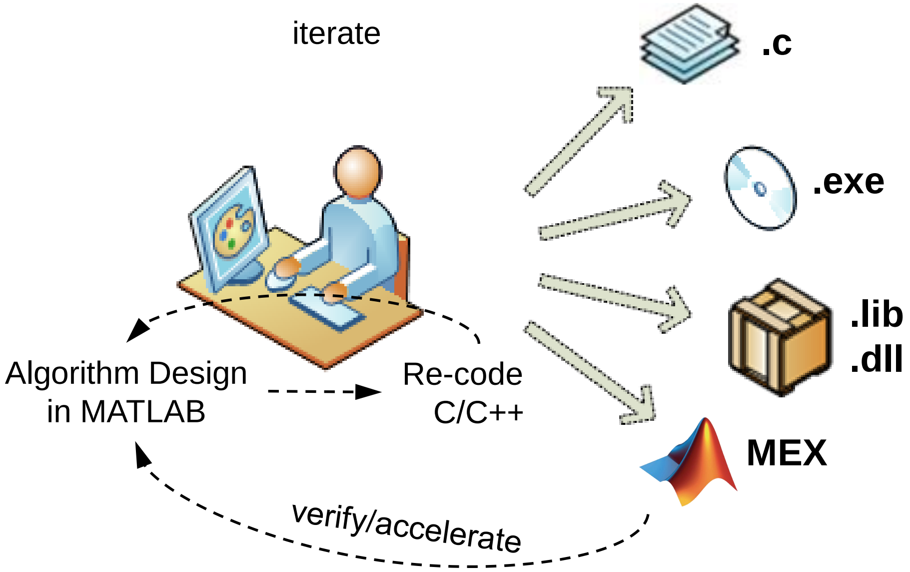
- Optimizing `for` loops and using vector and matrix operations
- Finding and addressing bottlenecks
- Generating C code and incorporating it into your application
- Utilizing additional hardware and processing power
- Summary and resources

Why Engineers Translate MATLAB to C

- Implement C code on processors or hand off to software engineers — .c
- Integrate MATLAB algorithms within existing C environments — .lib .dll
- Prototype MATLAB algorithms as standalone executables — .exe
- Accelerate MATLAB algorithms — MEX

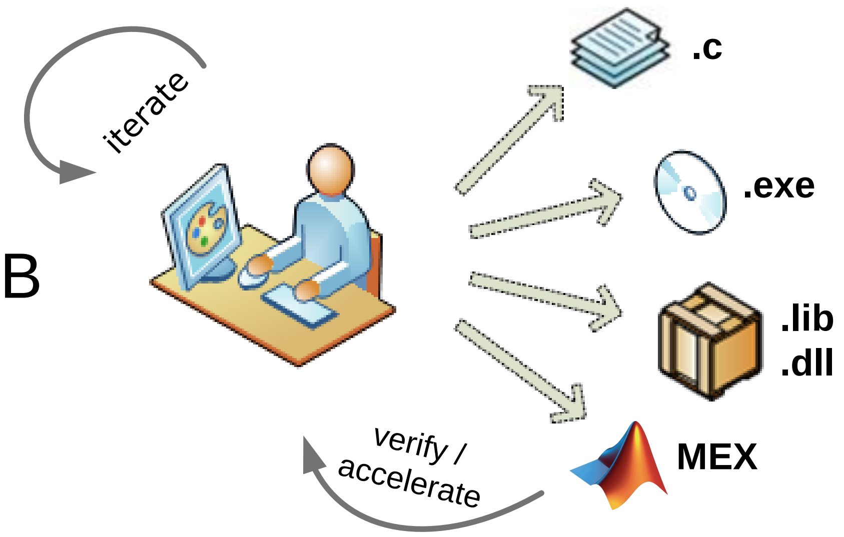
Challenges with Manual Translation of MATLAB to C

- Separate functional and implementation specifications
  - Leads to multiple implementations which are inconsistent
  - Hard to modify requirements during development
  - Difficult to keep MATLAB code and C code in sync
- Manual coding errors
- Time consuming and expensive process



Automatic Translation of MATLAB to C

- Maintain one design in MATLAB
- Design faster and get to C quickly
- Test more systematically and frequently
- Spend more time improving algorithms in MATLAB



Acceleration Using MEX

- Speedup factor will vary
- When you **may** see a speedup:
  - Often for communications or signal processing
  - Likely for loops with states or when vectorization is not possible
  - Always for fixed point
- When you **may not** see a speedup:
  - MATLAB implicitly multithreads computation
  - Built in functions that call BLAS or IPP

Supported Language Features and Functions

- New functions and features are supported each release

Matrices and Arrays	Data Types	Programming Constructs	Functions

• Matrix operations
• N-dimensional arrays
• Subscripting
• Frames
• Persistent variables
• Global variables
	• Complex numbers
	• Integer math
	• Double/single-precision
	• Fixed-point arithmetic
	• Characters
	• Structures
	• Cell arrays
	• Numeric class
	• Variable-sized data
	• MATLAB Class
	• System objects
		• Arithmetic, relational, and logical operators
		• Program control (`if`, `for`, `while`, `switch`)
			• MATLAB functions and subfunctions
			• Variable-length argument lists
			• Function handles
			
			Supported algorithms
			• More than 1100 MATLAB operators (R2015b), functions, and System objects for:
			  • Communications
			  • Computer vision
			  • Image processing
			  • Phased Array signal processing
			  • Robotics System Toolbox
			  • Signal processing
			  • Statistic & Machine Learning Toolbox

http://www.mathworks.com/help/coder/language-supported-for-code-generation.html

More Resources

- Product Page:
  - http://www.mathworks.com/products/matlab-coder
- On demand webinar, “MATLAB to C Made Easy”:
  - http://www.mathworks.com/videos/matlab-to-c-made-easy-81870.html

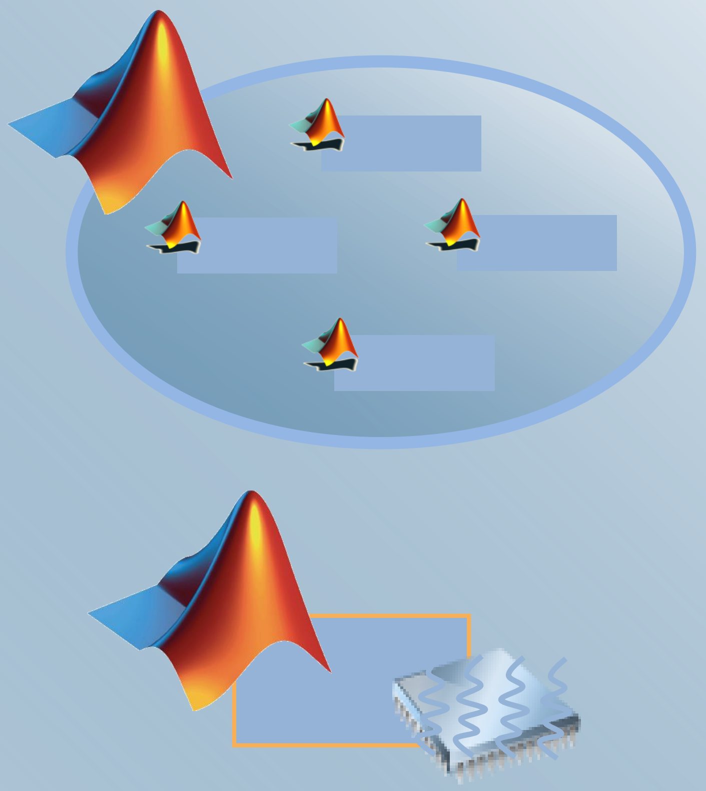
Agenda

- Optimizing `for` loops and using vector and matrix operations
- Finding and addressing bottlenecks
- Generating C code and incorporating it into your application
- Utilizing additional hardware and processing power
- Summary and resources

Parallel Computing enables you to…

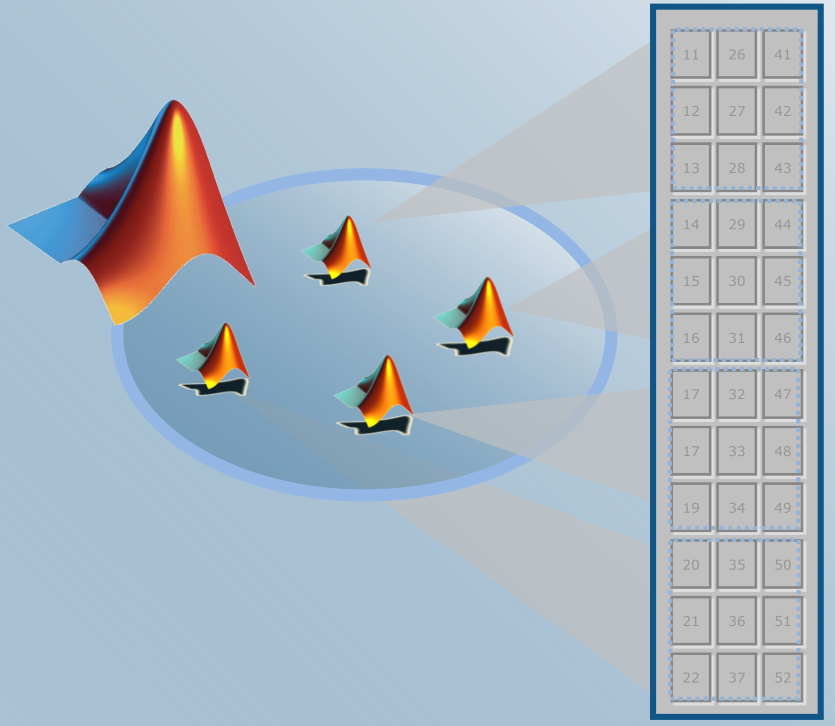
Larger Compute Pool

Speed up Computations

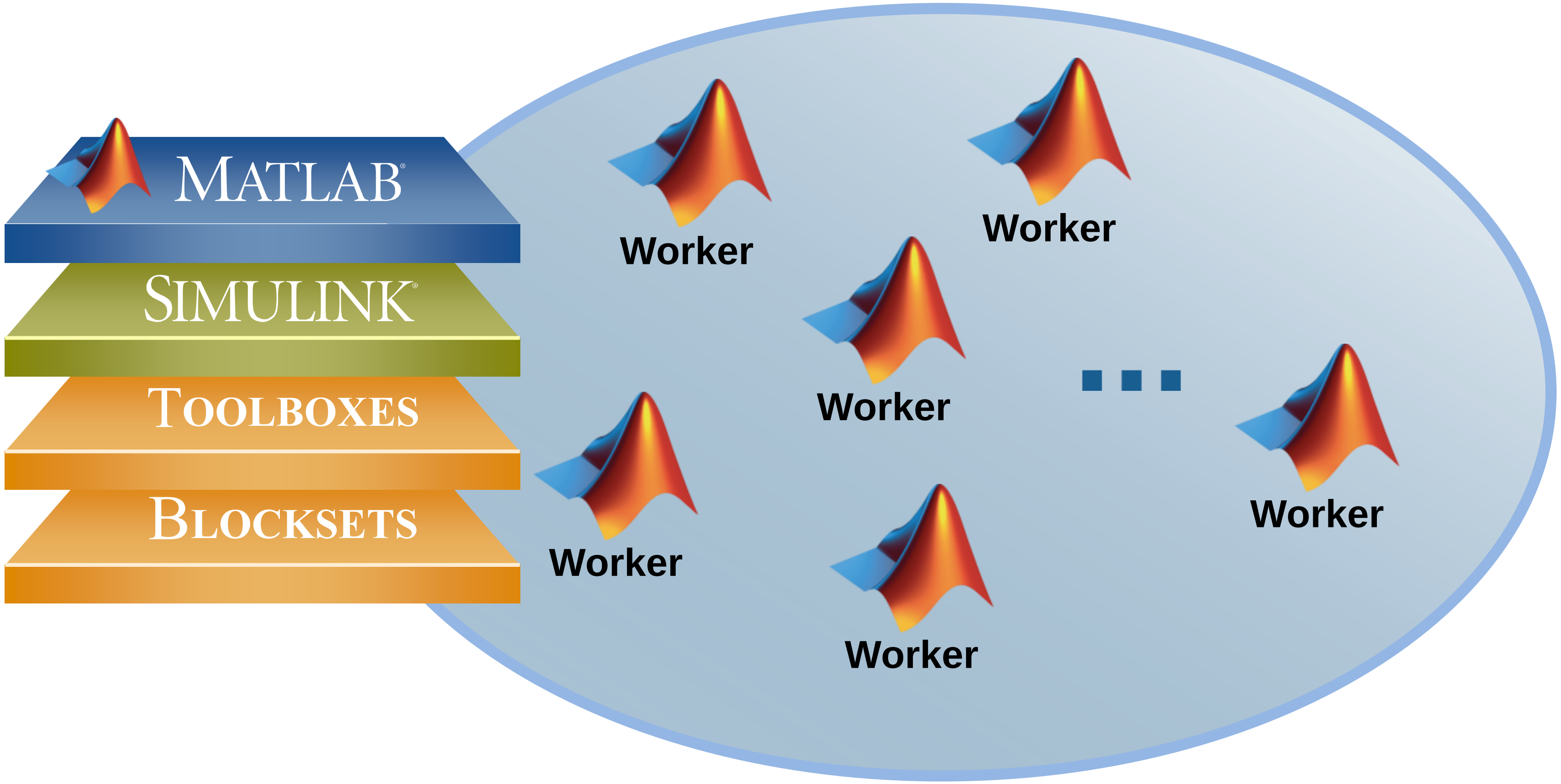


Larger Memory Pool

Work with Large Data

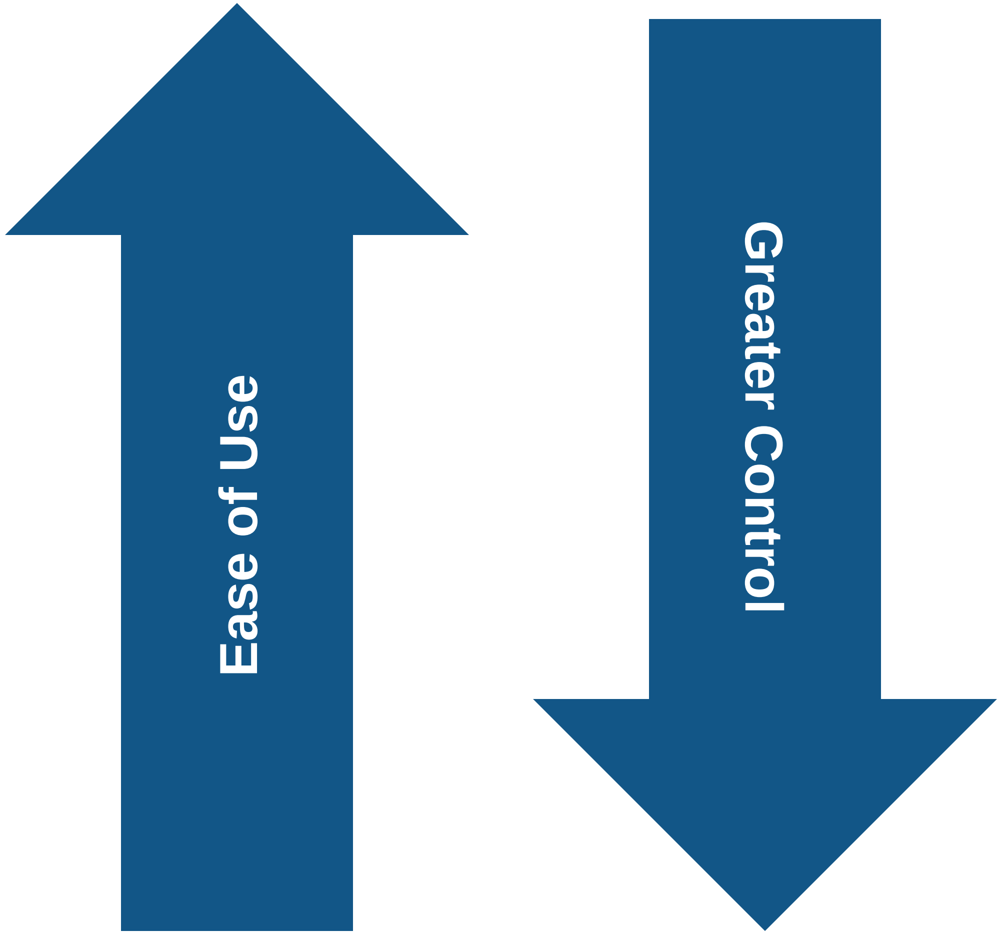


Parallel Computing with MATLAB



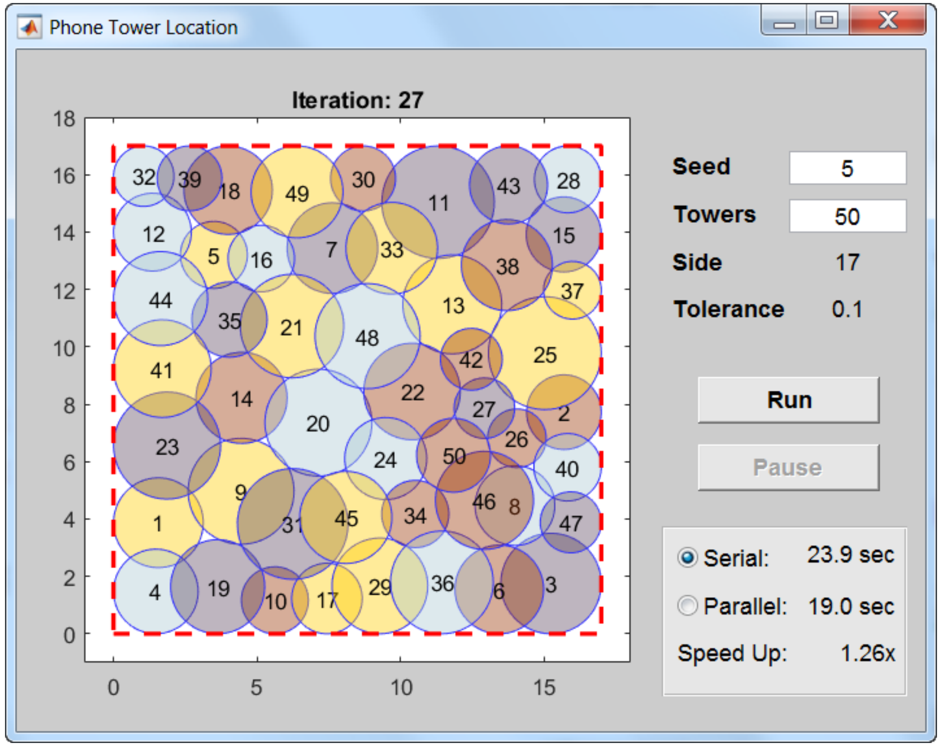
Programming Parallel Applications

- Built in support
  - `..., 'UseParallel', true)`



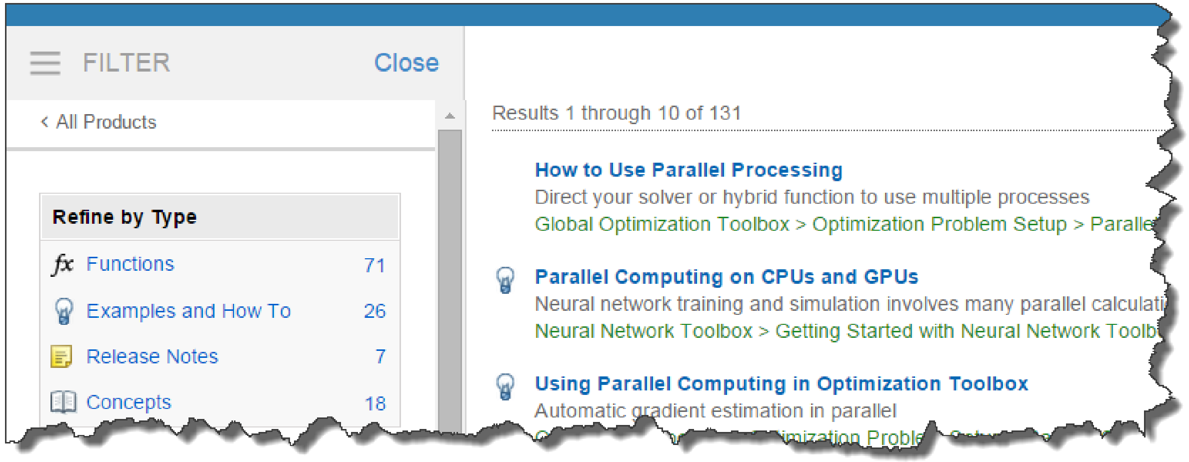
Example: Cell Phone Tower Optimization

- Run optimization with and without parallel
- Run different problem sizes



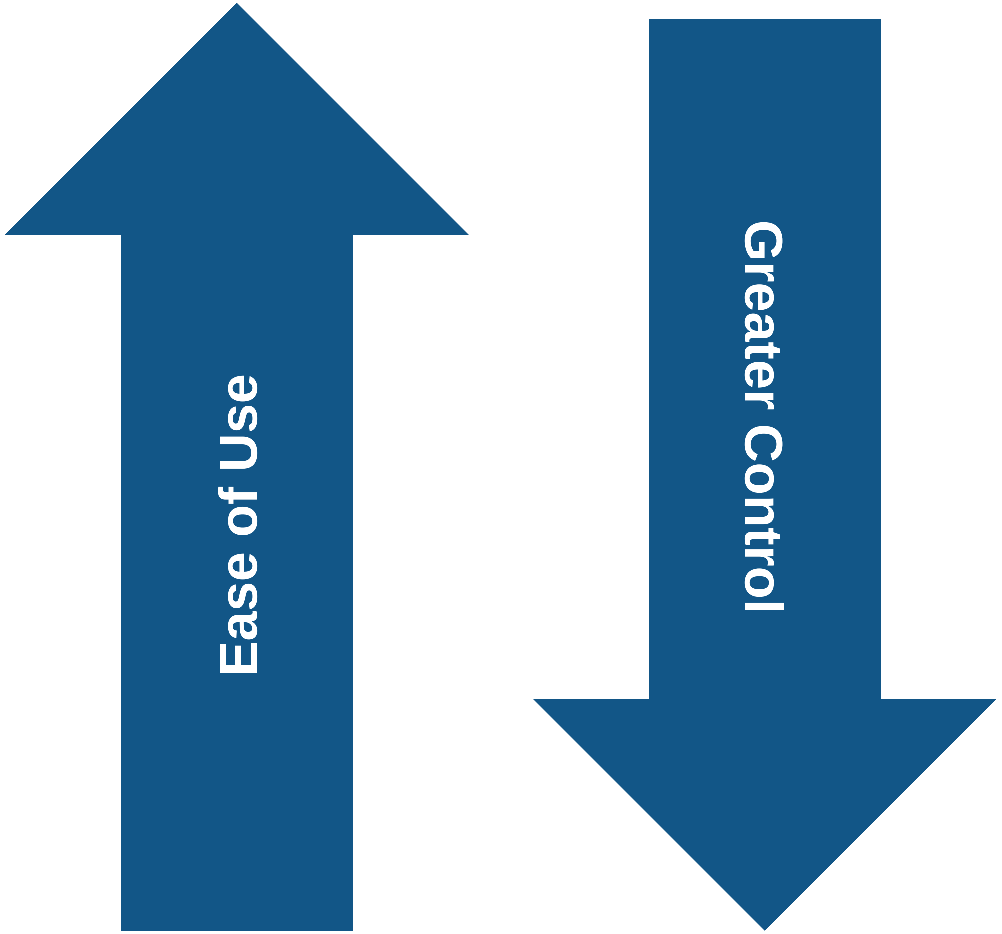
Products Providing Parallel Support

- Math, Statistics, Optimization
- Image Processing, Signal Processing, and Computer Vision
- Control System Design and Analysis
- Computational Biology
- Code Generation



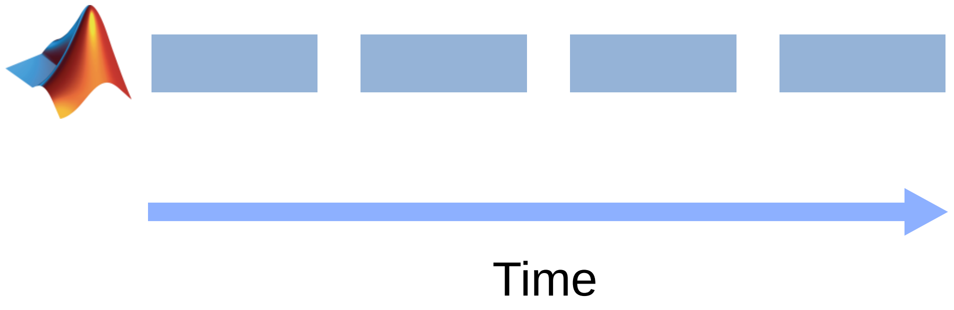
Programming Parallel Applications

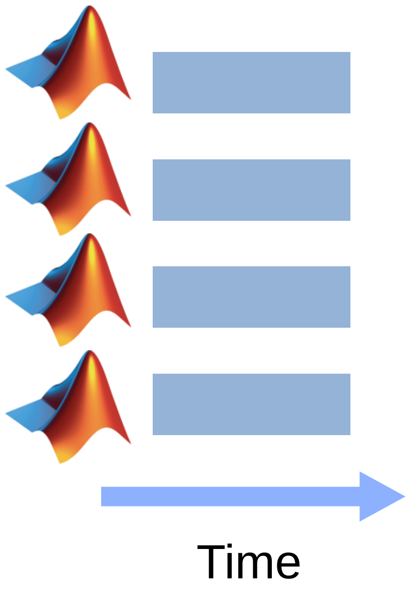
- Built in support
  - ..., 'UseParallel', true)
- Simple programming constructs
  - parfor, batch



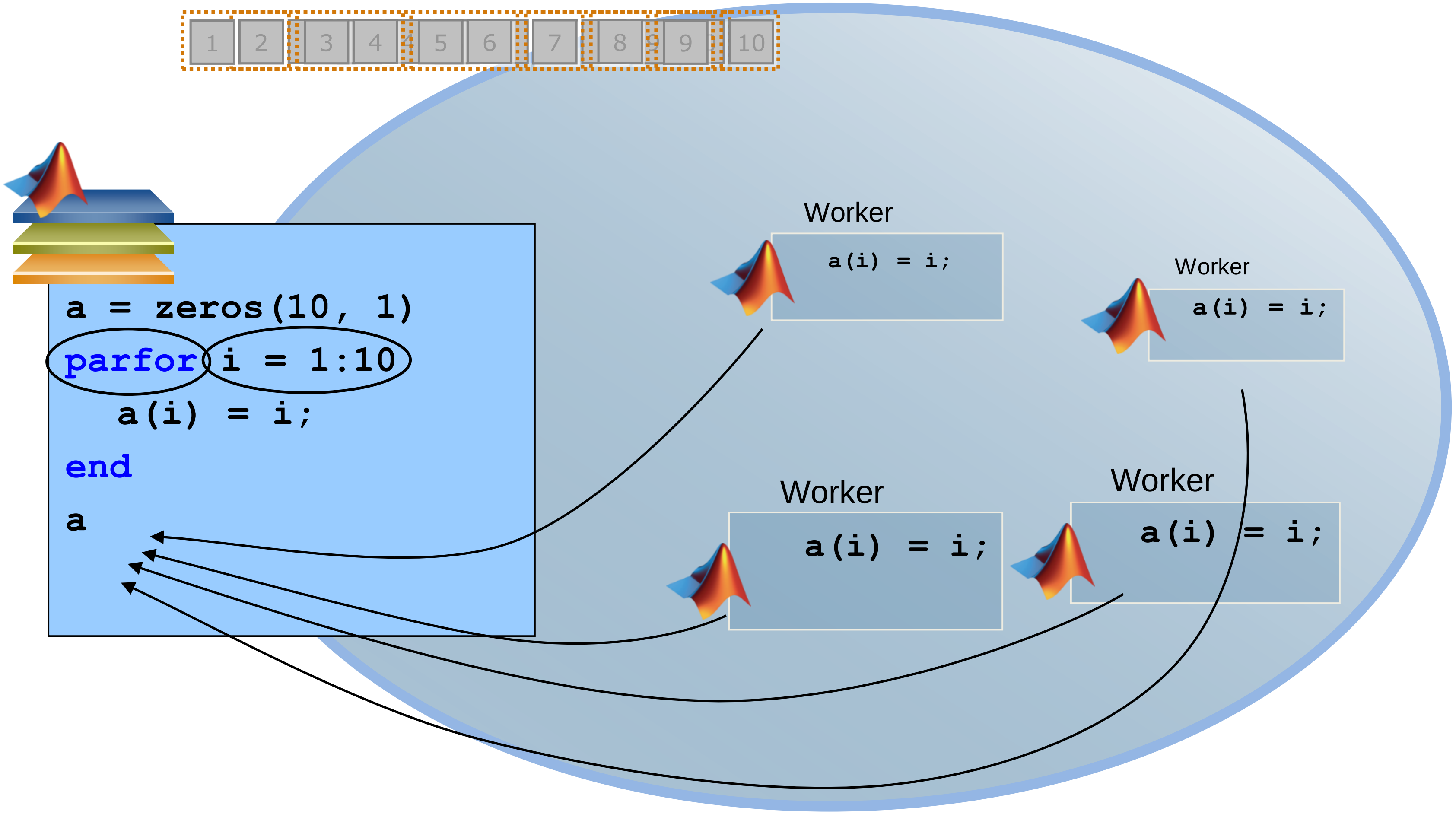
Embarrassingly Parallel Tasks

- No dependencies or communication between tasks
- Examples:
  - Monte Carlo simulations
  - Parameter sweeps
  - Same operation on many files



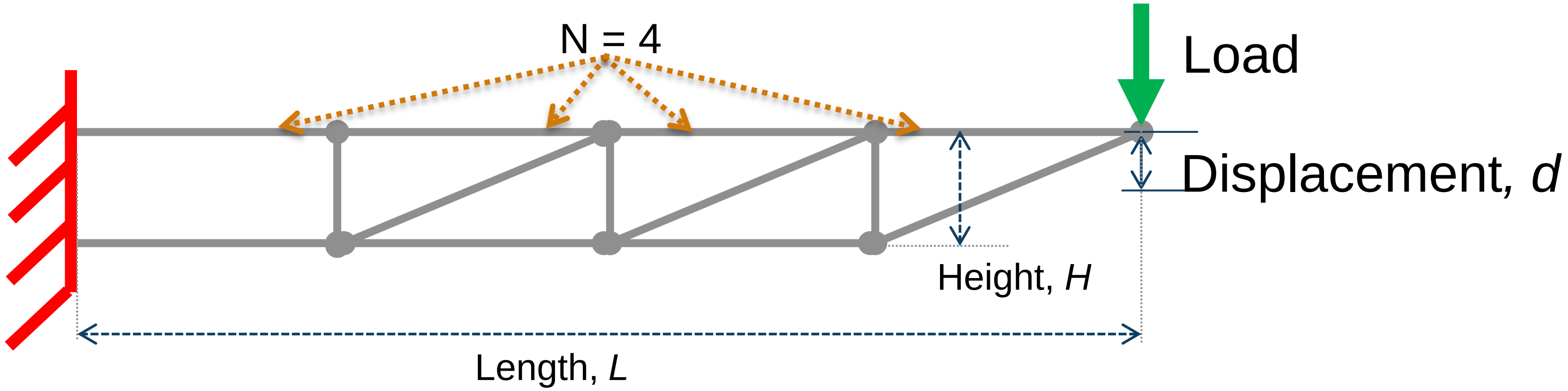


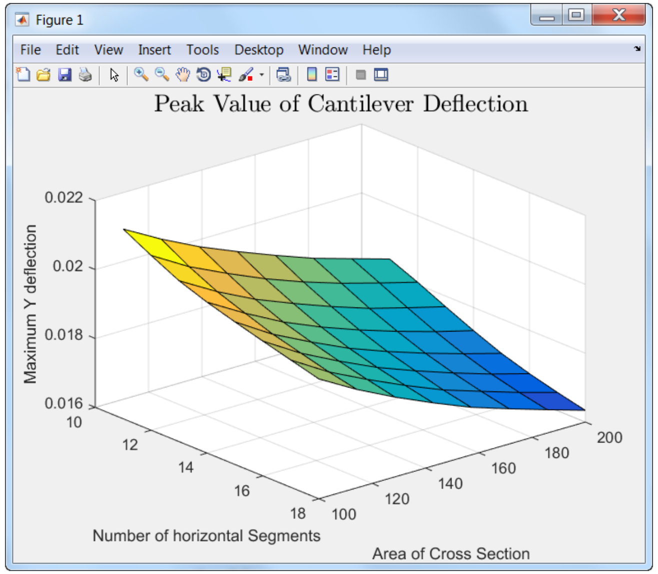
Mechanics of `parfor` Loops



Example: Parameter Sweep

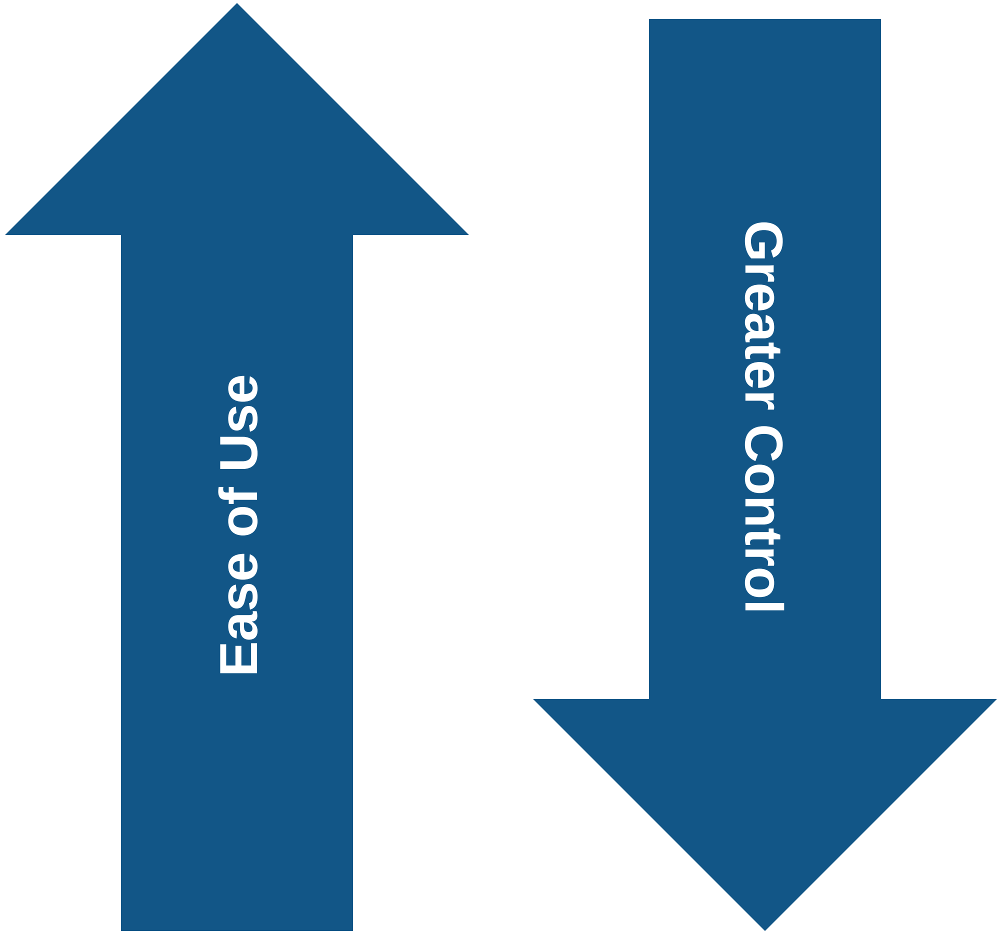
- Parameter sweep
  - Truss under a dynamic load
  - Sweeping over cross sectional area and number of elements





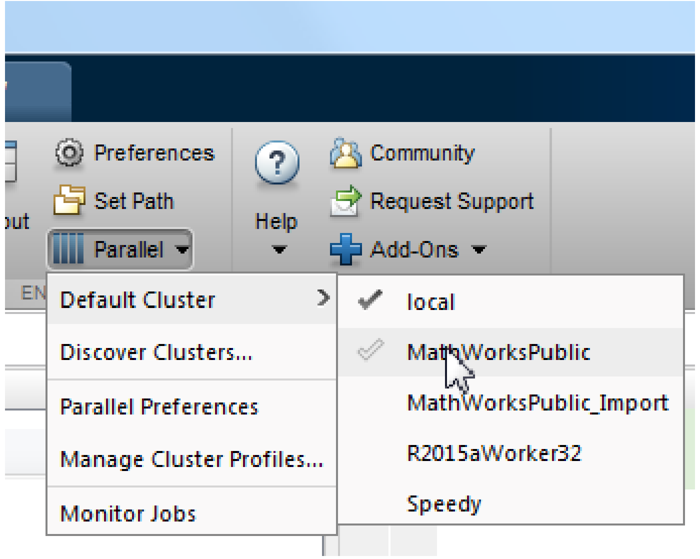
Programming Parallel Applications

- Built in support
  - ``..., 'UseParallel', true)``
- Simple programming constructs
  - `parfor, batch`
- Full control of parallelization
  - `spmd, parfeval`



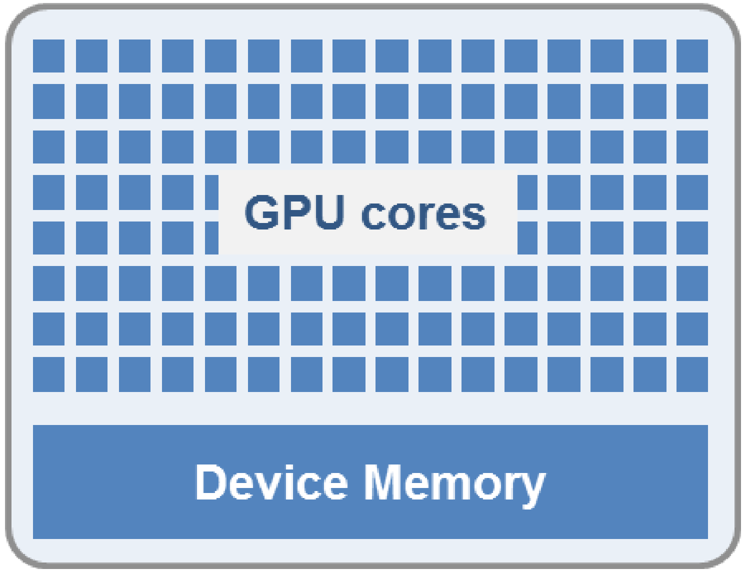
Migrate to Cluster / Cloud

- Use MATLAB Distributed Computing Server
- Change hardware without changing algorithm



Graphics Processing Units (GPUs)

- For graphics acceleration and scientific computing
- Many parallel processors
- Dedicated high speed memory



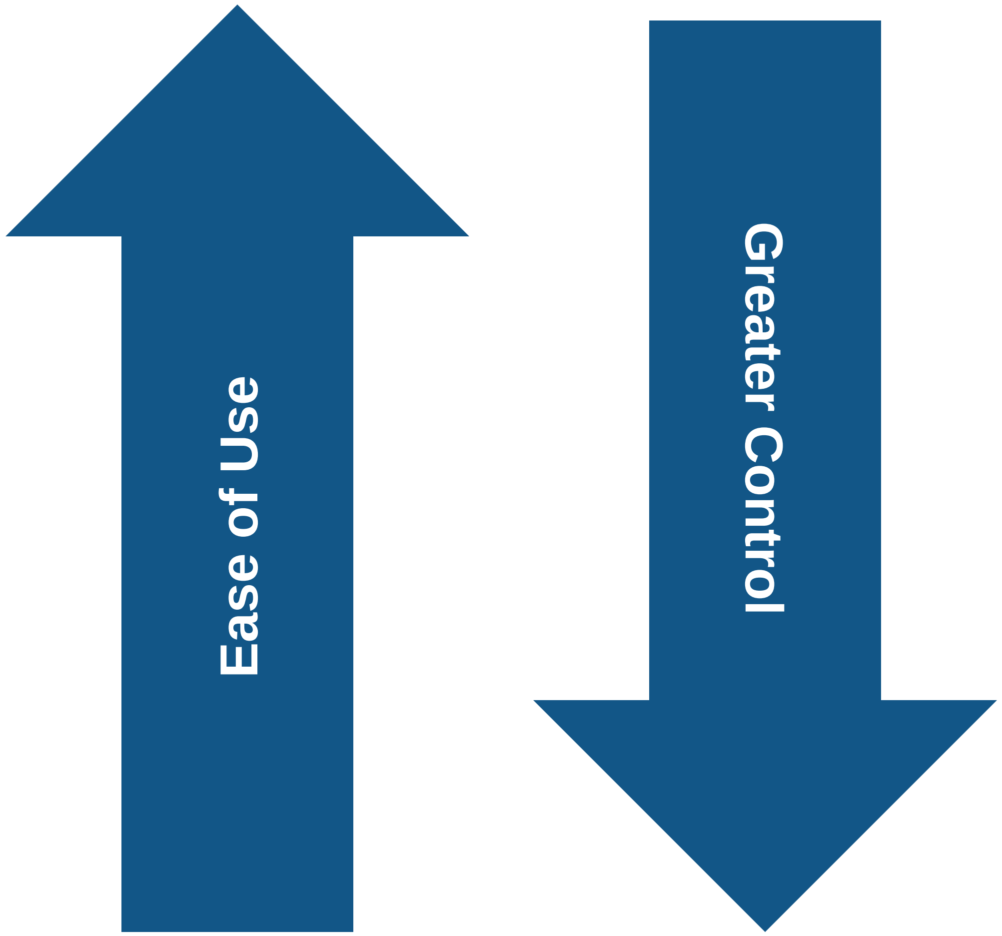
GPU Requirements

- Parallel Computing Toolbox requires NVIDIA GPUs
- www.nvidia.com/object/cuda_gpus.html

MATLAB Release	Required Compute Capability
MATLAB R2014b and newer releases	2.0 or greater
MATLAB R2014a and earlier releases	1.3 or greater

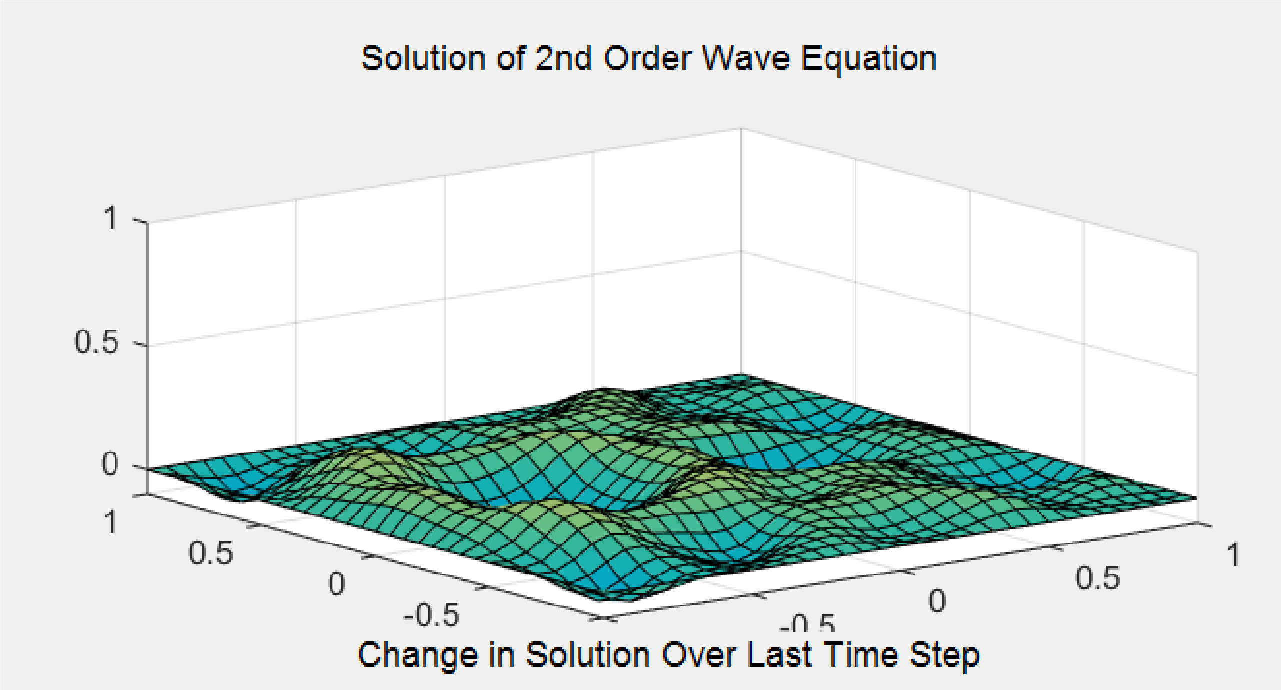
Programming with GPUs

- Built in toolbox support
- Simple programming constructs
  - `gpuArray, gather`

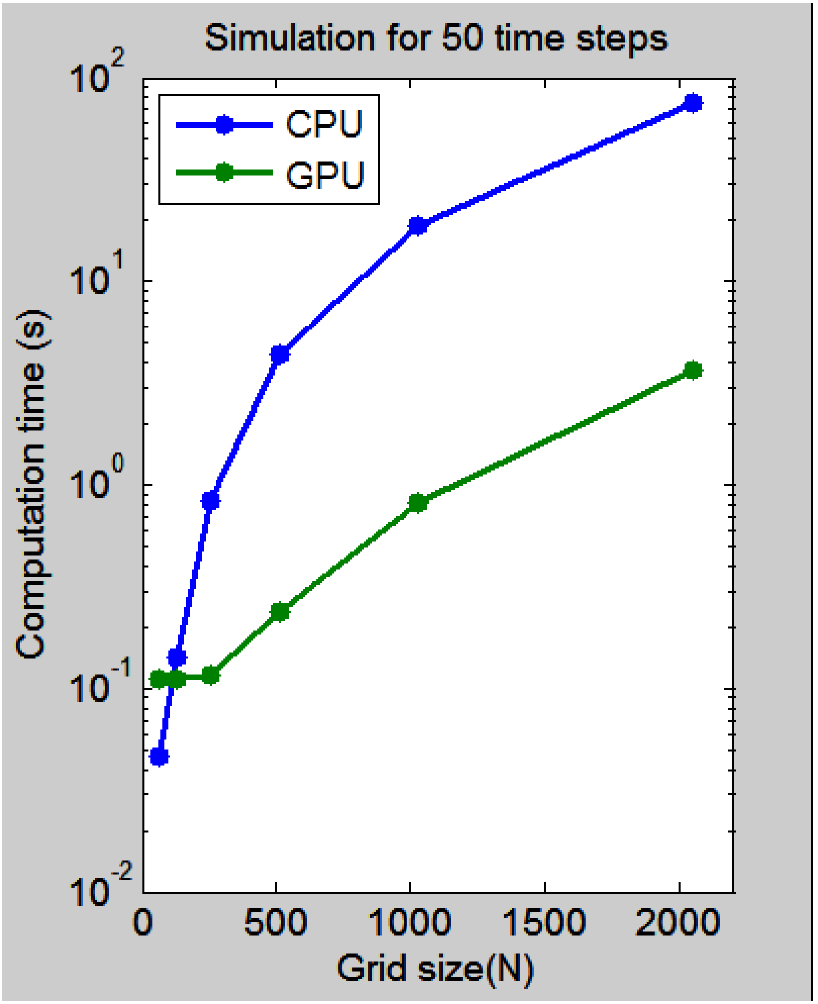


Example: Wave Equation

- Solve 2nd order wave equation with spectral methods
- Use CPU and GPU



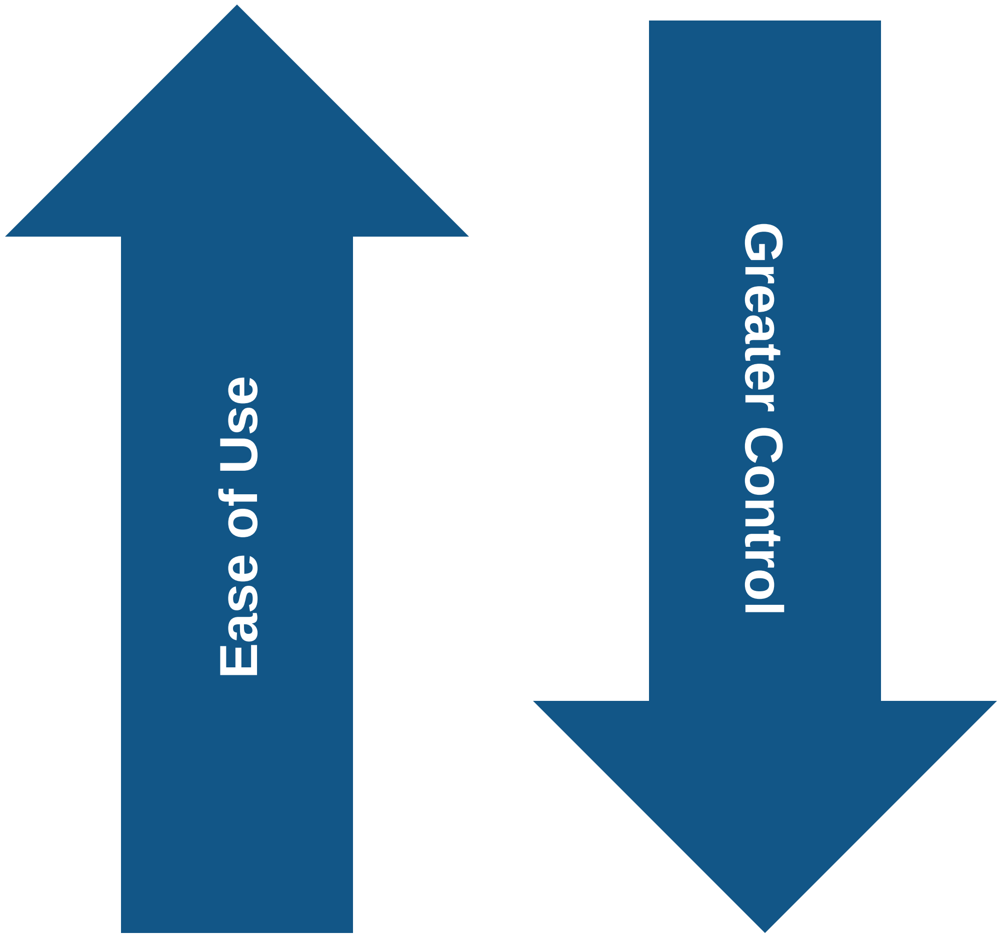
Benchmark: Solving 2D Wave Equation – CPU vs GPU



Intel Xeon Processor W3550 (3.07GHz), NVIDIA Tesla K20c GPU

Programming with GPUs

- Built in toolbox support
- Simple programming constructs
  - `gpuArray`, `gather`
- Advanced programming constructs
  - `spmd`, `arrayfun`
- Interface for experts
  - `CUDAKernel`, `mex`



Agenda

- Optimizing `for` loops and using vector and matrix operations
- Finding and addressing bottlenecks
- Generating C code and incorporating it into your application
- Utilizing additional hardware and processing power
- Summary and resources

Key Takeaways

- Consider the performance benefits of vector and matrix operations
- Analyze your code for bottlenecks to address the critical areas
- Leverage MATLAB Coder to speed up functions with generated C code
- Leverage parallel computing tools to take advantage of additional hardware

Some Other Valuable Resources

- MATLAB Documentation
  – MATLAB → Advanced Software Development → Performance and Memory
- Accelerating MATLAB algorithms and applications
  – http://www.mathworks.com/company/newsletters/articles/accelerating-matlab-algorithms-and-applications.html
- Loren Shure's Blog: "The Art of MATLAB"
  – http://blogs.mathworks.com/loren/
- MATLAB Question and Answers Site: MATLAB Answers
  – http://www.mathworks.com/matlabcentral/answers/