Efficient System Level Simulations using MATLAB and Simulink

Marc Willerton, Application Engineer
Email: marc.willerton@mathworks.co.uk

MathWorks at a glance

- ~3000 Employees worldwide with headquarters near Boston, USA
- Continued investment in new technologies
- A global company supporting global customers

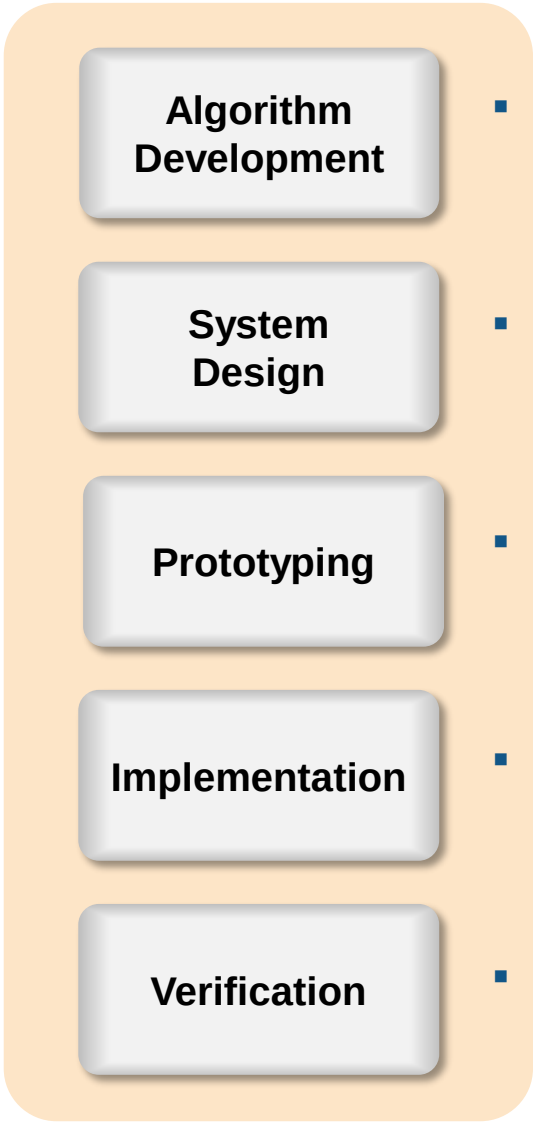
MathWorks UK

- ~160 Employees across 2 offices in Cambridge and Glasgow (80% Engineering)
- Cambridge office founded in 1984
  - MathWorks subsidiary in 1997
- Glasgow office founded in 2004
  - MathWorks subsidiary in 2013
- Continued investment in high-calibre engineering capability

MathWorks and Academia

- Research focussed seminars
- Developer talks, demonstrations and advisories
- Software Carpentry Workshops
- Sponsorship of Doctoral Training Centres
- Letters of support
- Sponsorship of PhD Students and Internships

MATLAB and Simulink in the UK Defence Industry



- MATLAB for designing algorithms, analysing real-world data
- Modelling complex dynamic systems, architectures of algorithms
- Rapid implementation on COTS real-time hardware
- Production quality coding on uP/DSP/FPGA platforms
- Verification processes to ensure quality and support certification

How do we take our algorithms further?

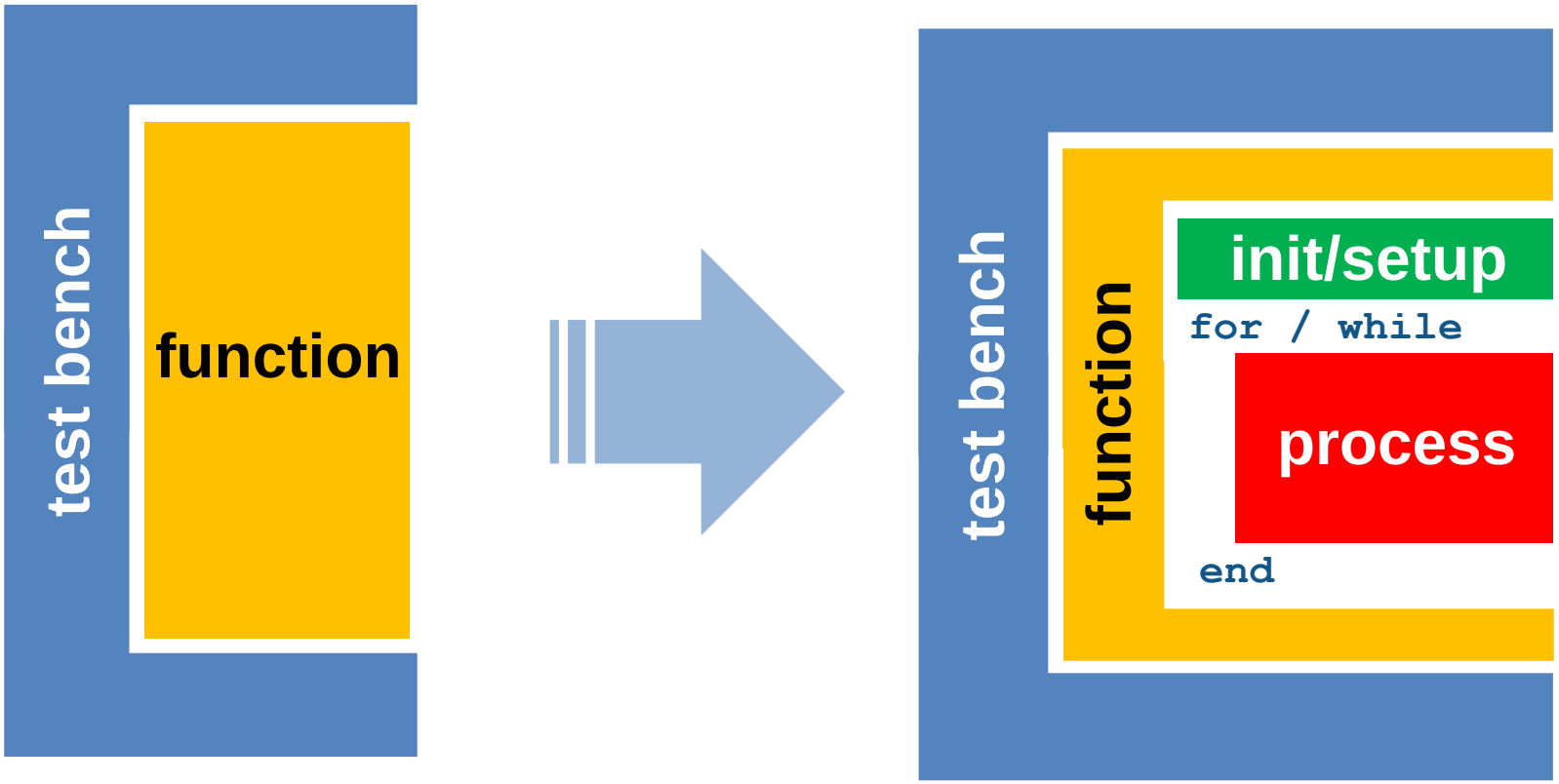
- UDRC developing research in Signal Processing
  - Performance analysis
  - Produce research papers
  - Provide input to UK Defence industry for future systems
- What else our technology can do to support this work
  - Faster simulations?
  - Interfacing with real-world data?
  - Rapid prototype/demonstrator development?
  - Better transfer of technology into industrial designs?

How do we take our algorithms further?

- Architect / review / optimize MATLAB code
- Accelerate using C/C++ code generation
- Run on parallel architectures (GPU / multi-core)

Refactoring MATLAB Code

- For example, separate **initialization and setup** from **recurring execution**
- Use profiler to analyse the performance improvement



Translating algorithms to C – Typical use cases

Hand-off code to software engineers for implementation (e.g. on embedded processor)

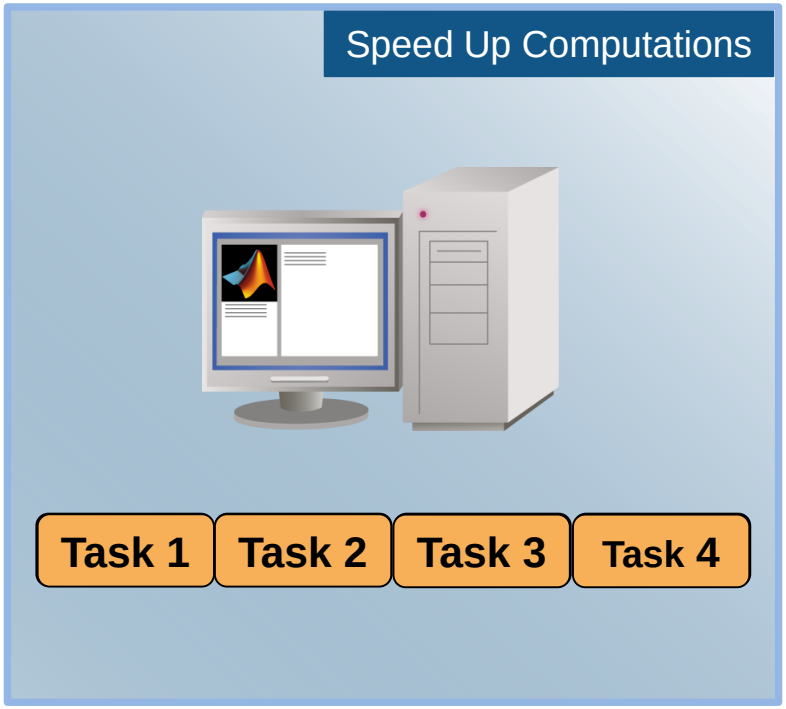
Integrate algorithms w/ existing C environment

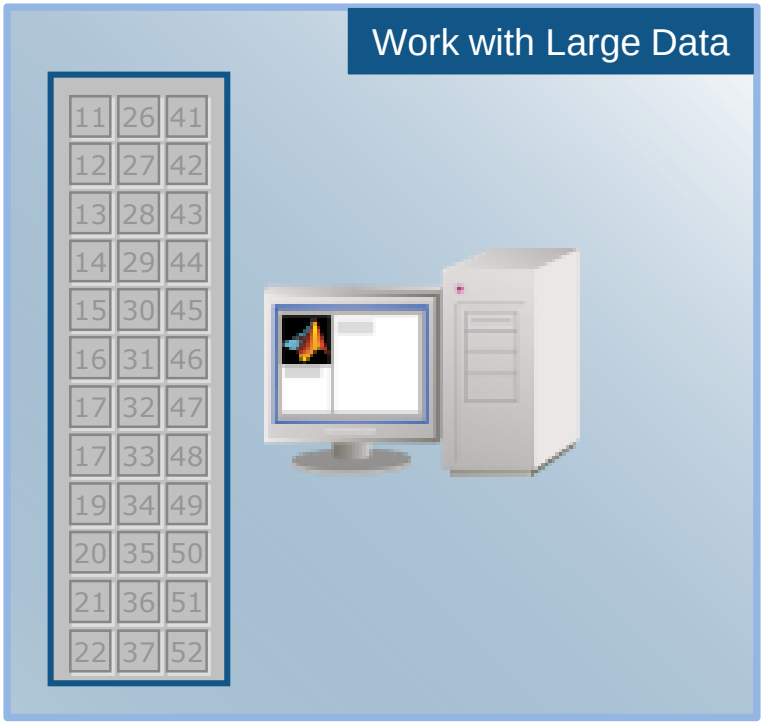
Deploy algorithms on Windows/Linux desktop PC

Accelerate algorithms

How Parallel Computing Toolbox may help

- Speeding up computations by distributing jobs to multiple MATLAB workers (e.g. using `parfor`)
- Distributing the memory burden through distributed arrays





About Graphics Processing Units (GPUs)

- Originally for graphics acceleration, now also used for scientific calculations
- Massively parallel array of integer and floating point processors
  - Typically hundreds of processors per card
  - GPU cores complement CPU cores
- GPU support introduced R2010b

Requires NVIDIA GPUs with Compute Capability 1.3 or greater, including NVIDIA Tesla 10-series and 20-series products.

Moving further towards deployment…

- Desktop Prototyping using Hardware Peripherals
- Deployment to Low Cost Prototyping Platforms
- Fixed Point Conversion
- Deploying to processors, FPGAs and SOCs
- Hardware-In-Loop (HIL) Verification

Questions?