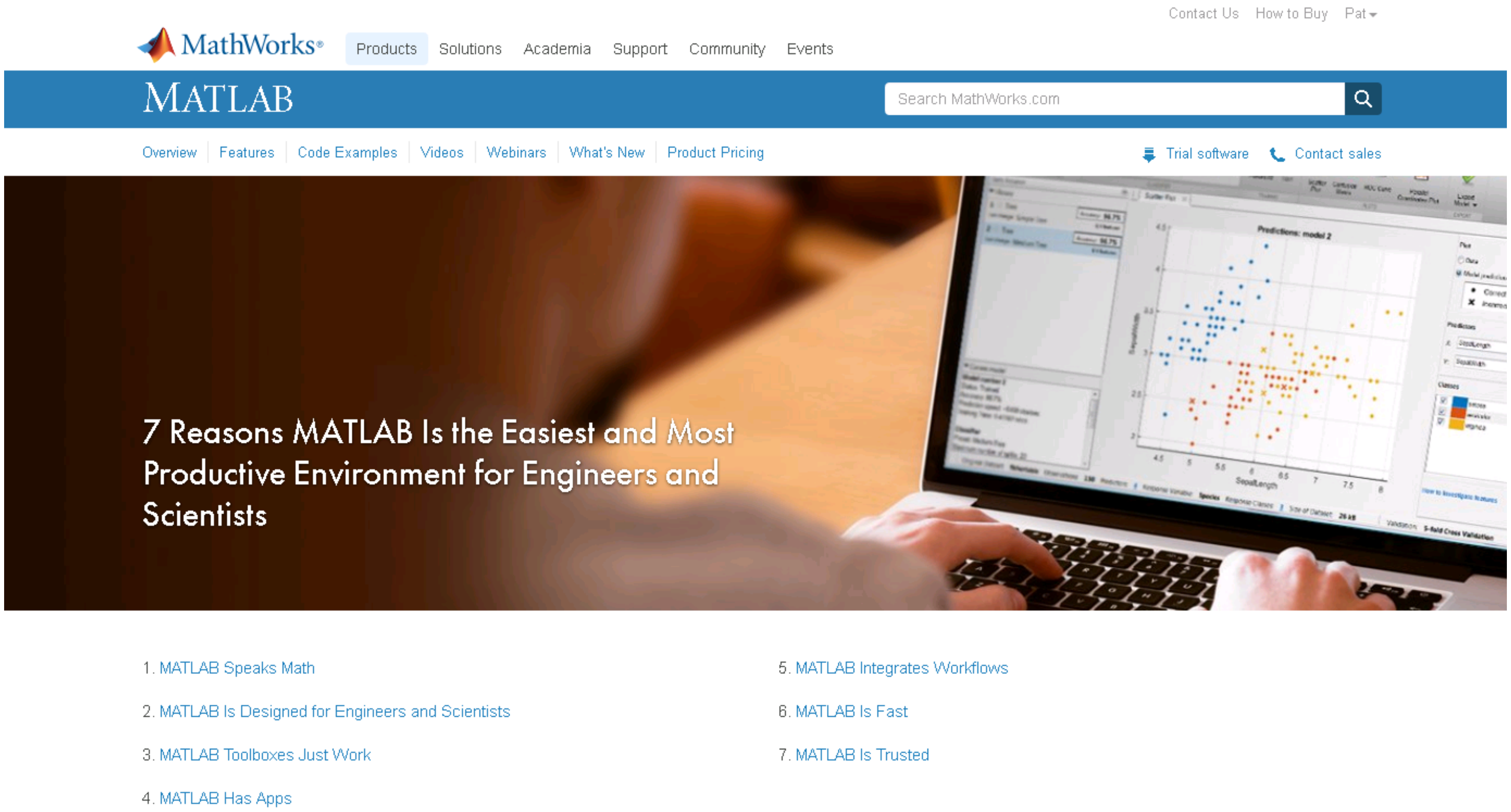
# MATLAB and BLAS

Pat Quillen, MATLAB Math

**Workshop on Batched, Reproducible, and Reduced Precision BLAS**

25 February 2017

# MATLAB Goals and Values



7 Reasons MATLAB Is the Easiest and Most Productive Environment for Engineers and Scientists

1. MATLAB Speaks Math
2. MATLAB Is Designed for Engineers and Scientists
3. MATLAB Toolboxes Just Work
4. MATLAB Has Apps
5. MATLAB Integrates Workflows
6. MATLAB Is Fast
7. MATLAB Is Trusted

# MATLAB and BLAS

- MATLAB has used BLAS from day 1.
- Expose BLAS directly to customers
  - `C = A*B;`
  - `B = A\X; % Triangular A`
- Incorporate BLAS in 3p libraries we use
- Leverage BLAS in our own built-ins
  - Stick to BLAS standard names

# Configuring MATLAB BLAS

- Can set environment variables to point to BLAS (and LAPACK) of your choice
  - `BLAS_VERSION`, `BLAS_VERBOSITY`
- Link against `libmwblas.so` to use, e.g. through mex
- We’ll tell you which BLAS we’re using
  - `version -blas`

# Batched BLAS at MathWorks

- Little used currently
- Biggest impact likely to be made by use within 3p libraries
- `gpuArray` in parallel toolbox offers `pagefun`
  - Limited to subset of functions
  - Equivalent to fixed batch
  - Customer exposure requires another data structure?

# Exposing to Customers

- So far, not a lot of customer demand
- Enable current syntaxes to offer batched functionality?
  - `C = A*B;   % A is m-by-n-by-p, B is n-by-k-by-p`
  - `X = A\B;   % A is m-by-n-by-p, B is m-by-k-by-p`
- `pagefun` is the most likely route
- 16b and Implicit Expansion
  - For elementwise functions, dimensions must match or be 1
  - `A – mean(A)`
  - Actual Customer feedback:  “What an idiotic decision!”

# Reproducibility for MathWorks

- Guarantee the same results in successive runs with
  - Same data (including alignment, ordering)
  - Same hardware
  - Same OS and version
  - Same version of MATLAB and underlying libraries
  - Same number of threads
- Generally, prefer usability, productivity over performance
- Are we thinking big enough?

# Reproducibility for our dependencies

- Good: Can be used in some limited way to get the reproducibility we need
- Better: Can be configured to be reproducible
  - Control use randomness in algorithms
  - Control threading
  - Control arithmetic (e.g MKL CNR control)
- Best: Are themselves reproducible

# Exceptions and non-finites

- Generally, `NaN` in, `NaN` out*
- Don’t (yet) care about particular `NaN`s
  - But don’t want to gratuitously change patterns
- Some years ago we removed warnings for
  - `log(0)`
  - `x/0`
- Minimal user complaints
  - Do BLAS functions really need an `INFO` return?

# Reduced/Mixed Precision BLAS

- Reduced Precision
  - Little to no customer demand for MATLAB
  - Hear about it for GPU products especially for Deep Learning
  - Possible interest in MATLAB→Simulink→Embedded targets
- Mixed Precision
  - Little explicit demand
  - How to expose to customers if we wanted to?
- Mixed Real/Complex (same precision) GEMM would be nice

# Summary

- MATLAB values correctness, usability, productivity, performance
  - Reproducibility is key
  - Should we insist on more?
- Puzzles around how to expose concepts and power of Batched BLAS to customers
  - Is it necessary to expose top-level API to customers?
  - Good enough to leverage where we can?
- Want MATLAB to be a platform that can enable experimentation
  - Through mex, or sufficient use of `BLAS_VERSION`, `LAPACK_VERSION`
- Happy to be part of this workshop!