What’s New in MATLAB

Joe Hicklin
MATLAB as a Collaboration Platform

MathWorks Talk

MATLAB as a Collaboration Platform

Application Track 1 at 12:15 by Marta Wilczkowiak

Explore Today
Big Data in MATLAB

Memory and Data Access
- 64-bit processors
- Memory Mapped Variables
- Disk Variables
- Databases
- Data stores – **NEW in R2014b**

Platforms
- Desktop (Multicore, GPU)
- Clusters
- Cloud Computing (MDCS on EC2)
- Hadoop – **NEW in R2014b**

**MathWorks Talk**

Data Analytics with MATLAB

Masterclass Track at 15:45 Adrienne James

- Block Processing
- Parallel-for loops
- GPU Arrays
- SPMD and Distributed Arrays
- MATLAB MapReduce – **NEW in R2014b**
What’s New in Optimization?

- Mixed Integer Linear Programming solver (*intlinprog*) **R2014a**
  - Plot functions and output functions for monitoring progress of *intlinprog* solver **R2014b**

- Particle Swarm Optimization (*particleswarm*) **R2014b**

**MathWorks Talk**

Best Practices for Optimisation in MATLAB

*Application Track 2 at 12:15 Jasmina Lazić*
This Talk:

- The new graphics system
- The new datetime tools
- Talking to an Android Phone or Tablet
This Talk:

- The new graphics system
- The new datetime tools
- Talking to an Android Phone or Tablet
This Talk:

- The new graphics system
- The new datetime tools
- Talking to an Android Phone or Tablet
This Talk:

- The new graphics system
- The new datetime tools
- Talking to an Android Phone or Tablet
MATLAB Graphics

- New implementation.
- Backwards compatible.
- Many Advantages
Higher Quality

- We fixed a large number of bugs

- Modular architecture
  - continued high quality
  - more features in the future
The new graphics look better than before.

We want to increase your ability to see your data.

Data Colors,
Line widths,
Fonts and Font sizes,
Axis colors,
Grid colors …
The new graphics look better than before.

- New architecture allows us to take more advantage of graphics cards.
  - Line smoothing, also called anti-aliasing eliminates “pixel-jagged” lines

![Bessel Functions](image_url)
Can anyone name this handsome bird?

Tropical Parula
Our new color map is named Parula.
Graphics objects are now really objects

```matlab
>> p = plot(Y)

p =

174.0016

>>
```
Graphics objects are now really objects

```matlab
>> p = plot(Y)

p =

Line with properties:

    Color: [0 0.4470 0.7410]
   LineStyle: '-'
    LineWidth: 0.5000
    Marker: 'none'
    MarkerSize: 6
    MarkerFaceColor: 'none'
    XData: [1 2 3 4 5 6 7 8 9 10 11 12]
    YData: [1x12 double]
    ZData: []
```

Show all properties
>> Setting properties with “dot notation”

- `set(p,'LineWidth',4)`
- `p.LineWidth = 4`
- `set(p,'LineWidth',get(p,'LineWidth') + 3)`
- `p.Color = 'red'`
- `p.Marker = 'square'`
Many new features you have been asking for

- Separate color maps on individual axes
- Rotatable tick labels
- Tabs in your GUIs
- Localization / internationalization
- Animated line for better animations
Many new features you have been asking for

- Separate color maps on individual axes
Many new features you have been asking for

- Rotatable tick labels

```matlab
>> myAxes.XTickLabelRotation = 45
```
Many new features you have been asking for

- Tabs in your GUIs

```
% Create group of tabs
tabGroup = uitabgroup;

% Add tabs
tab1 = uitab(tabGroup,'Title','Loan Data');
tab2 = uitab(tabGroup,'Title','Amortization Table');
tab3 = uitab(tabGroup,'Title','Principal/Interest Plot');

% Set location of tabs
tabGroup.TabLocation = 'top';
```
Many new features you have been asking for

- Localization / internationalization
Many new features you have been asking for

- Animated line for better animations
  - Erasemode = ‘none’ is gone.
  - You tell us ahead of time that you plan to add data to this line.
  - We draw smarter…
Try out the new graphics system

- It makes better charts of your data
- It is easier to use
- It is more powerful
datetime and duration

- Three new objects
  - datetime
  - duration
  - calendarDuration

- The old way has problems…
  - Numbers,
  - Strings,
  - Date vectors
datetime and duration
datetime and duration

A datetime represents a point in time

```matlab
>> t = datetime('now')
```

t =

```
7-Oct-2014 09:50:09
```
datetime and duration

A duration represents an interval of time

```
>> d = 2 * hours
    d =
    2 hrs
```
datetime and duration

A duration is NOT located in time

These are all the same duration.
datetime and duration

When you add two durations…

Time

>> d1 = 2 * hours
  d1 =
    2 hrs
>> d2 = 234 * minutes
  d2 =
    234 mins
When you add two durations…

$$d_1 = 2 \times \text{hours}$$
$$d_1 = 2 \text{ hrs}$$
$$d_2 = 234 \times \text{minutes}$$
$$d_2 = 234 \text{ mins}$$

$$d_1 + d_2$$
$$\text{ans} = 5.9 \text{ hrs}$$
datetime and duration

When you add a duration to a datetime...

```
>> T = datetime('now')
T =
    26-Sep-2014 13:43:26

>> D = 200 * hours
D =
    200 hrs

>> T + D
```
When you add a duration to a datetime...

You get a new datetime

>> T = datetime('now')
T =
    26-Sep-2014 13:43:26

>> D = 200 * hours
D =
    200 hrs

>> T + D
ans =
    04-Oct-2014 21:43:26
datetime and duration

When you subtract a datetime from a datetime…

>> T1 = datetime('now')
T1 =
26-Sep-2014 13:43:26

>> T2 = datetime('now')
T2 =
26-Sep-2014 13:47:46
>> T2 - T1
datetime and duration

When you subtract a datetime from a datetime…

>> T1 = datetime('now')
T1 =
    26-Sep-2014 13:43:26

>> T2 = datetime('now')
T2 =
    26-Sep-2014 13:47:46

>> T2 - T1
ans =
    00:04:20

You get a duration
When you add a datetime to a datetime…

\[
\text{T1 = datetime('now')}
\]
\[
\text{T1 = 26-Sep-2014 13:43:26}
\]

\[
\text{T2 = datetime('now')}
\]
\[
\text{T2 = 26-Sep-2014 13:47:46}
\]

\[
\text{T1 + T2}
\]
datetime and duration

When you add a datetime to a datetime...

```
>> T1 = datetime('now')
T1 =
 26-Sep-2014 13:43:26

>> T2 = datetime('now')
T2 =
 26-Sep-2014 13:47:46

>> T1 + T2
Addition is not defined between datetime arrays.
```
datetime and duration

How many seconds in an hour?

1 hr

Time
datetime and duration

How many seconds in a week?

1 week

Time
datetime and duration

How many seconds in a month?

1 month

Time
datetime and duration

How many seconds in a month?

1 month

Time

Which month?
Which year?
Leap second?
Daylight Savings Time?
**calendarDuration**

Like a duration, BUT, exact length depends on “Which”

Use “cal*” functions
- calDays
- calWeeks
- calMonths
- calQuarters
- calYears
datetime and duration

When you add a calendarDuration to a datetime...

\[
>> T = \text{datetime('now')}
T =
01-Oct-2014 15:43:11
\]

\[
>> D = 2 * \text{calmonths}
D =
2mo
\]

\[
>> T + D
\]
**datetime and duration**

When you add a `calendarDuration` to a `datetime`…

```
>> T = datetime('now')
T =
 01-Oct-2014 15:43:11

>> D = 2 * calmonths
D =
  2mo

>> T + D
ans =
 01-Dec-2014 15:43:11
```
Plotting with datetime

>> plot(time, temperature)
Plotting with datetime

```matlab
gg plot(time, temperature)
```
Plotting with datetime

>> plot(time, temperature)
datetime

- `d.TimeZone`
- Combining from different time zones

- The datetime object does reading, math, and display/formatting all in one integrated package. People have struggled with that
datetime

- You can get or set each part of a datetime.

```matlab
>> d = datetime('now')
d = 02-Oct-2014 15:11:59

>> d.Day
ans =
    2

>> d.Hour
ans =
    15

>> d.Month = 4
d = 02-Apr-2014 15:11:59
```
datetime

datetimes can be configured to read and display in a dizzying range of formats.

```matlab
>> d = datetime('now')

d =
    02-Oct-2014 15:02:52

>> d.Format = 'eeee QQQQ, yyyy h:mm'

d =
    Thursday 4th quarter, 2014 3:02
datetime

Vectors of datetimes

>> datetime(2014,3,16:20)

ans =

datetimes can optionally have time zones.

```matlab
>> d1 = datetime(2014,3,16,17,30,0,'TimeZone','GMT')
d1 = 16-Mar-2014 17:30:00

>> d2 = datetime(2014,3,16,17,30,0,'TimeZone','EST')
d2 = 16-Mar-2014 17:30:00

>> d3 = [d1;d2]
d3 =
16-Mar-2014 17:30:00
16-Mar-2014 22:30:00

>> d3.Format = 'H:M:S zzzz'
d3 =
17:3:0 UTC
22:3:0 UTC
```
Try them out!

- `datetime`
- `duration`
- `calendarDuration`
MATLAB can now talk to your Android device!

- MATLAB talks to lots of hardware
  - Arduino
  - Lego Mind Storm
  - Robot Operating System
  - Raspberry Pi
  - Thing Speak
  - USB Webcams
  - IPhone
  - Lots more…
MATLAB can now talk to your Android device!

- MATLAB talks to lots of hardware
  - Arduino
  - Lego Mind Storm
  - Robot Operating System
  - Raspberry Pi
  - Thing Speak
  - USB Webcams
  - IPhone
  - Lots more…

- And now, Android!
Four steps to set up

1. Get the Android support package
2. Get MATLAB Mobile
   Requires Android 4.0 or greater, aka Ice Cream Sandwich, Jelly Bean, or KitKat
3. Turn it on in MATLAB
4. Turn it on on your phone
Android support package

www.mathworks.com/hardware-support/android-sensor.html
Android support package
Android support package

Android Sensor Support from MATLAB
Use MATLAB to acquire accelerometer, magnetometer, orientation, and location data from built-in sensors on your Android device.

MATLAB® supports the acquisition of data from built-in sensors on your Android™ device. With the MATLAB Support Package for Android Sensors, you can log data or query the most recent data available from the supported sensors on your Android device. These include motion sensors like the accelerometer and position sensors like the GPS. Measurements such as acceleration, magnetic field, latitude, longitude, and altitude can be viewed on your Android smartphone and tablet. You can then analyze the data in MATLAB® or build applications that make decisions based on the sensor data. To use this package, you must also install MATLAB Mobile™ on your Android device.

The MATLAB Support Package for Android Sensors lets you acquire and log data from the supported Android sensors to obtain the indicated measurements:

- Acceleration on 3-axes
- Magnetic field on 3-axis
- Angular velocity on 3-axes
- Azimuth, roll, pitch
- Latitude, longitude, altitude, horizontal accuracy, speed, and course

Prior Release Support
For Android Sensor support for MATLAB versions R2014a or earlier see: Android Sensor Support for R2014a and earlier

Platform Support
Available on 32-bit and 64-bit Microsoft Windows, 64-bit Mac OS X, and 64-bit Linux

Code Examples

- Getting Started Resources
- Examples
- Documentation
- Solutions
- MathWorks Requirements
- Third-Party Requirements

Ready to install?
Get Support Package Now
What's Next?
Get MATLAB Mobile from Google play
Configure your phone or tablet to send the data you want.

- Acceleration on 3-axes
- Angular Velocity on 3-axes
- Magnetic Field on 3-axes
- Orientation (Azimuth, Pitch and Roll)
- Position
  - Latitude, Longitude, Altitude
- Horizontal Accuracy
- Speed
- Course
Read the data

>> myPhone = mobiledev

myPhone =
mobiledev with properties:

    Connected: 1
    Logging: 0
    InitialTimestamp: ''

    AccelerationSensorEnabled: 0
    AngularVelocitySensorEnabled: 0
    MagneticSensorEnabled: 0
    OrientationSensorEnabled: 0
    PositionSensorEnabled: 0
Read the data

```matlab
>> myPhone.AngularVelocitySensorEnabled = 1;
>> myPhone.Logging = 1;

>> [acceleration, time] = myPhone.accellog;

>> plot(time, acceleration)
```
That’s it!

- New graphics system
- New datetime tools
- Talking to an Android Phone or Tablet

- Questions?
  - Now
  - Any time today.