

MATLAB*P Quick Start Guide

Ron Choy, MIT CSAIL
Modified for UCSB by Viral Shah

April 1, 2004

1 Running MATLAB*P

Use *getresv* to reserve processors. Then run the script *startmp* to start MATLAB*P. Say you want to start with 8 processes:

```
$ getresv 4  
$ startmp
```

will give you a MATLAB session ready for use with MATLAB*P. Here is what the startup looks like:

```
[viral@head viral]$ getresv 4  
qsub: waiting for job 64.head.cs.ucsb.edu to start  
qsub: job 64.head.cs.ucsb.edu ready
```

```
[viral@node004 viral]$ startmp
```

```
LAM 7.0.4/MPI 2 C++/ROMIO - Indiana University
```

```
Starting MATLAB*P  
MATLAB*P started, IP=172.16.0.4, port=33124  
Starting MATLAB with no X support
```

```
< M A T L A B >  
Copyright 1984-2002 The MathWorks, Inc.  
Version 6.5.0.180913a Release 13  
Jun 18 2002
```

Using Toolbox Path Cache. Type ‘‘help toolbox_path_cache’’ for more info.

To get started, type one of these: *helpwin*, *helpdesk*, or *demo*.
For product information, visit www.mathworks.com.

Connecting to MATLAB*P Server with 8 processes

>>

2 MATLAB*P Basics

The use of MATLAB*P can best be illustrated with a sample session (with annotation):

```
>> % check to see if server is alive, and the number of processes running
>> pping
ans =
     4
```

```
>> % create a 100x100 column-distributed random dense matrix
>> A = randn(100,100*p)
A =
```

```
      ddense object: 100-by-100
```

```
>> % retrieve the first element
>> A(1,1)
ans =
    0.3855
```

```
>> % create a 100x100 row-distributed random dense matrix
>> B = randn(100*p,100);
```

```
>> % solve the system AX=B
>> X = A\B;
```

```
>> % check the answer
>> norm(A*X-B)
ans =
    2.7259e-13
```

```
>> % get information about variables
>> whose
```

Your variables are:

Name	Size	Bytes	Class
A	100x100p	80000	ddense array
B	100px100	80000	ddense array

```
X          100px100  80000    ddense array
ans        1x1      8         double array
```

```
Grand total is 30001 elements using 240008 bytes
MATLAB has a total of 1 elements using 8 bytes
MATLAB*P server has a total of 30000 elements using 240000 bytes
```

```
>> % get the time on all processes
>> times = mm('clock');
>> times
```

```
times =
```

```
          ddense object: 4-by-6
>> times(:, :)
ans =
  1.0e+03 *
  2.0030    0.0100    0.0070    0.0210    0.0250    0.0303
  2.0030    0.0100    0.0070    0.0210    0.0250    0.0303
  2.0030    0.0100    0.0070    0.0210    0.0250    0.0303
  2.0030    0.0100    0.0070    0.0210    0.0250    0.0303
```

```
>> % create a distributed vector containing the matrix ranks
>> i = 1:np*p;
```

```
>> % compute pi using adaptive quadrature in 'MultiMATLAB' mode
>> my_pi = mm('quad', '4./(1+x.^2)', (i-1)/np, i/np)
```

```
my_pi =
```

```
          ddense object: 1-by-4
>> sum(my_pi)
```

```
ans =
```

```
3.1416
```

3 List of functions and brief explanation

3.1 Functions for distributed dense matrices

Operators

& | ~ ==
< > <= >=
~= - Matrix logical and comparison operators
, .' - Transpose and conjugate transpose
a+b - Binary addition
a-b - Binary subtraction
-a - Unary minus
a.*b - Element-wise multiplication
a*b - Matrix multiplication
a./b - Right element-wise division
a.\b - Left element-wise division
a/b - Matrix right division
a\b - Matrix left division
a.^b - Element-wise power
a^b - Matrix power
horzcat
vertcat - Horizontal/vertical concatenation
subsasgn - Subscripted assignment
subsref - Subscripted reference

Trigonometric functions

sin sinh
cos cosh
tan tanh - Trigonometric functions

Others

abs - Absolute value
ceil - Round towards plus infinity
chol - Cholesky factorization
cov - Covariance
diag - Diagonal matrix
dot - Dot product
eig - Eigenvalue
exp - Elementwise exponential
find - Find indices of nonzero elements
floor - Round towards minus infinity
hess - Hessenberg form
inv - Matrix inverse
isreallyreal - Checks if matrix is of real storage
isreal - Checks if matrix has no imaginary elements

issym	- Checks if matrix is symmetric
isvalid	- Checks if matrix is a valid distributed matrix
length	- Length of vector
log10	- Element-wise log base 10
log	- Element-wise log
lu	- LU factorization
max	- Largest component
mean	- Average or mean value
min	- Smallest component
nnz	- Number of nonzeros
norm	- Matrix/vector norm
round	- Round towards nearest integer
size	- Size of matrix
sqrt	- Element-wise square root
sum	- Sum of elements
svd	- Full singular value decomposition
tril	- Extract lower triangular part
triu	- Extract upper triangular part

3.2 General functions

mm	- 'MultiMATLAB mode'
np	- Returns number of processes
p	- Constructs parallel matrices (see help)
pp2matlab	- Transfers a matrix from MATLAB*P server to MATLAB
matlab2pp	- Transfers a matrix from MATLAB to MATLAB*P server
whose	- Returns information about variables. Analogous to whos in MATLAB.