Matlab Array Basics and Miscellaneous

back to Fan's Intro Math for Econ, Matlab Examples, or MEconTools Repositories

Check Parameter Types

There parameter input can either be a cell array or an integer, conditional processing based on parameter input type

```
% Float and Cell
curEstiParamA = 1;
curEstiParamB = {146, 'R3'};
% test if is float
st_test = strjoin(...
    ["", ...
     ['isfloat(curEstiParamA)=' num2str(isfloat(curEstiParamA))], ...
     ['isfloat(curEstiParamB)=' num2str(isfloat(curEstiParamB))], ...
    ], ";");
disp(st_test);
```

```
;isfloat(curEstiParamA)=1;isfloat(curEstiParamB)=0
```

```
% test if is cell
st_test = strjoin(...
    ["", ...
     ['iscell(curEstiParamA)=' num2str(iscell(curEstiParamA))], ...
     ['iscell(curEstiParamB)=' num2str(iscell(curEstiParamB))], ...
    ], ";");
disp(st_test);
```

```
;iscell(curEstiParamA)=0;iscell(curEstiParamB)=1
```

Compare Array Values That are Approximately Similar

What is the best way to compare floats for almost-equality in Python?

- rel_tol is a relative tolerance, it is multiplied by the greater of the magnitudes of the two arguments; as the values get larger, so does the allowed difference between them while still considering them equal.
- abs_tol is an absolute tolerance that is applied as-is in all cases. If the difference is less than either of those tolerances, the values are considered equal.

```
rel_tol=1e-09;
abs_tol=0.0;
if_is_close = @(a,b) (abs(a-b) <= max(rel_tol * max(abs(a), abs(b)), abs_tol));
disp(['1 and 1, if_is_close:' num2str(if_is_close(1,1))]);
```

```
1 and 1, if_is_close:1
```

```
disp(['1e-300 and 1e-301, if_is_close:' num2str(if_is_close(1e-300,1e-301))]);
```

```
1e-300 and 1e-301, if_is_close:0
```

```
disp(['1+1e-9 and 1+1e-10, if_is_close:' num2str(if_is_close(1+1e-9,1+1e-10))]);
```

```
1+1e-9 and 1+1e-10, if_is_close:1
```

Imaginary Number Examples

```
rng(123);

% Imaginary array
ar_img = rand([1,7]) + 1i*rand([1,7]);

% Regular Array
ar_real = rand([1,10]);

% Combine arrays
ar_full = [ar_real ar_img];
ar_full = ar_full(randperm(length(ar_full)));
disp(ar_full);
```

```
  Columns 1 through 7

   0.6344 + 0.0000i   0.1755 + 0.0000i   0.5316 + 0.0000i   0.2861 + 0.4809i   0.7380 + 0.0000i   0.1825 + 0.0000i

  Columns 8 through 14

   0.2269 + 0.3921i   0.7245 + 0.0000i   0.8494 + 0.0000i   0.6110 + 0.0000i   0.4231 + 0.4386i   0.9808 + 0.0597i

  Columns 15 through 17

   0.3980 + 0.0000i   0.5513 + 0.3432i   0.7195 + 0.7290i
```

```

% real index
disp(~imag(ar_full));
```

```
   1   1   1   0   1   1   0   0   1   1   1   0   0   1   1   0   0
```

```

% Get Real and not real Components
disp(ar_full(imag(ar_full) == 0));
```

```
    0.6344    0.1755    0.5316    0.7380    0.1825    0.7245    0.8494    0.6110    0.5318    0.3980
```

```
disp(ar_full(imag(ar_full) ~= 0));
```

```
   0.2861 + 0.4809i   0.6965 + 0.6848i   0.2269 + 0.3921i   0.4231 + 0.4386i   0.9808 + 0.0597i   0.5513 + 0.3432i
```