Lecture x: MATLAB - advanced use cases

Optimization

Heikki Apiola

April 16, 2019

Aalto University
juha.kuortti@aalto.fi, heikki.apiola@aalto.fi

Optimization

Why

Optimization methods provide us with a large number of examples that need lots of computational resources. Our interest here is to find ways for effcient computation. Our main goal is to see what parallel toolbox can do for us.

Goals of parallel computing

Repeat:

- **Efficiency**: Gain remarkable speedup distributing a computational task between several computing units, which work simultaneously with parts of the task. (Decompose your task into independent parts that may or may not communicate with each other.)
- **Memory**: Solve larger problems than can be done in one computational unit.

Matlab's min/max + spmd

Example: Find min/max of

$f(x) = x \sin x, \quad x \in [-\pi, n\pi]$

Decompose I into subintervals of length `L=(B-A)/nlabs` (`A=-pi, B=n*pi`).

Get this file (click) spmdOptim1D.m and

```
>> edit spmdOptim1D.m
```

Let's work together on it!

This exercise shows that especially in 1-dim cases, efficient and easy solutions can be obtained just by very basic vector operations.

Optimization in several variables

- Optimization toolbox, also Global OptT
- Various ways of studying parallelization
- Some uses of symbolic toolbox
- fmincon,fminsearch,lsqfit, ...

Unconstrained example 1

Find min and max of

$$f(x, y) = x \exp(-(x^2 + y^2)) + (x^2 + y^2)/20.$$

minmax2dsolver2.m

```
f = @(x,y) x.*exp(-x.^2-y.^2)+(x.^2+y.^2)/20;
fsurf(f,[-2,2],'ShowContours','on')
```

Practice vectorization, parfor and fminsearch,

Rosenbrock's banana-function

Classical test function for optimization routines:

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2$$

- Load this m-file into Matlab-editor: minmax2dsolRosenbrock.m
- Run the first few cells to get a visual idea.

Optimization steps are similar to the previous example, so let's leave them now.

Go on to constrained optimization with this objective function.

Nonlinear fitting

```
x = 20:65;
y = [0 0 0 1 2 5 15 65 71 79 55 48 46 26 25 25 16 9 ...
    18 ...
    8 8 6 4 6 5 5 2 6 4 2 0 0 1 1 1 0 1 1 0 0 0 0 0 ...
        2 0 0];

f = @(beta,x)beta(1)*x.^9 .* exp(-beta(2)*x);
fobj = @(lam)norm(f(lam,x)-y);
beta0 = [1 -1];
betaOpt = fminsearch(fobj,beta0);

xfit = linspace(20,65,128);
plot(xfit,f(betaOpt,xfit),x,y)
```

Traveling salesman

```
6.4 petaflops, 30 cities, brute force:
>> factorial(30)/(6.4*10^15)/(3600*24*360)
ans =
   1.3325e+09
```