Chapter 5 Project: Apply Nelder-Mead to the Rheology Problem

Matthew Saurette, Tyler Weames, and Sarah Wyse

Math 462
University of British Columbia - Okanagan

December 2020

Nelder-Mead Algorithm

Given $f : \mathbb{R}^n \mapsto \mathbb{R}$ and the vertices of an initial simplex $\mathrm{Y}^0 = \{\mathrm{y}^0, \mathrm{y}^1, \ldots, \mathrm{y}^n\}$

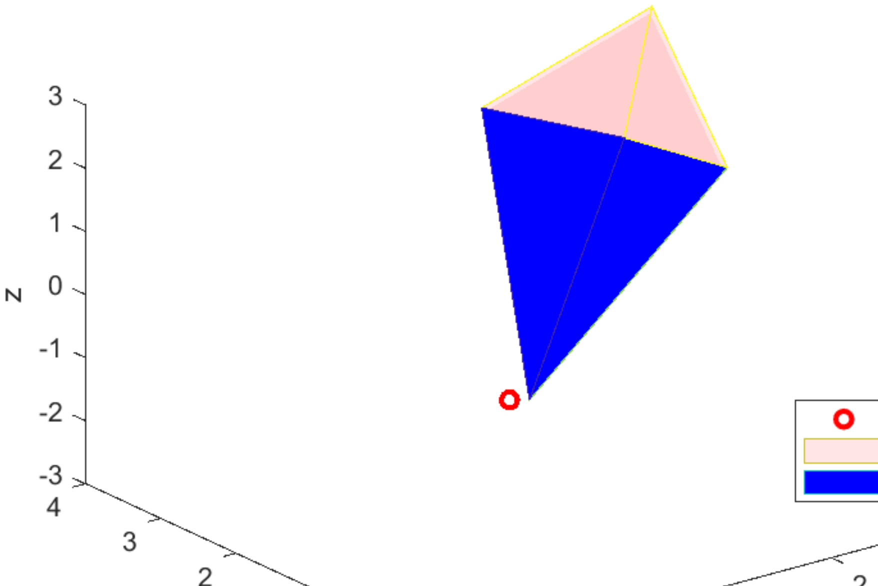
0. Initialize:
   - $\delta^e, \delta^{oc}, \delta^{ic}, \gamma$ parameters
   - $k \leftarrow 0$ iteration counter
1. Order and create centroid:
   - reorder Y^k so $f(y^0) \leq f(y^1) \leq \ldots \leq f(y^n)$
   - set $x^c = \frac{1}{n} \sum_{i=0}^{n-1} y^i$, the centroid of all except the worst point
2. Reflect:
   - test reflection point $x^r = x^c + (x^c - y^n)$
     - if $f(y^0) \leq f(x^r) < f(y^{n-1})$, then accept x^r and goto 1
3. Expand:
   - if $f(x^r) < f(y^0)$, then test expansion point $x^e = x^c + \delta^e(x^c - y^n)$

4a). Outside Contraction:
   - if $f(y^{n-1}) \leq f(x^r) < f(y^n)$, then test outside contraction $x^{oc} = x^c + \delta^{oc}(x^c - y^n)$

4b). Inside Contraction:
   - if $f(x^r) \geq f(y^n)$, then test inside contraction point $x^{ic} = x^c + \delta^{ic}(x^c - y^n)$

5. Shrink:
   - if all tests fail, then shrink
     $\mathrm{Y}^{k+1} = \{\mathrm{y}^0, \mathrm{y}^0 + \gamma(\mathrm{y}^1 - \mathrm{y}^0), \mathrm{y}^0 + \gamma(\mathrm{y}^2 - \mathrm{y}^0), \ldots, \mathrm{y}^0 + \gamma(\mathrm{y}^n - \mathrm{y}^0)\}$

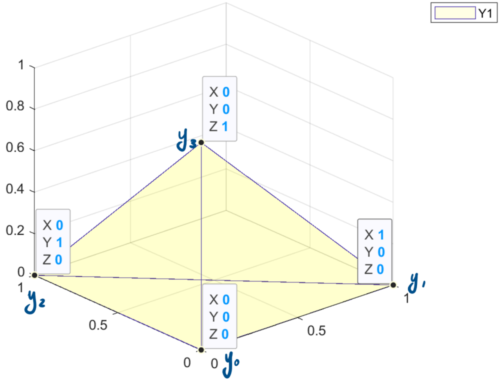
(Math 462, UBCO, 2020)



M

0. Initialize

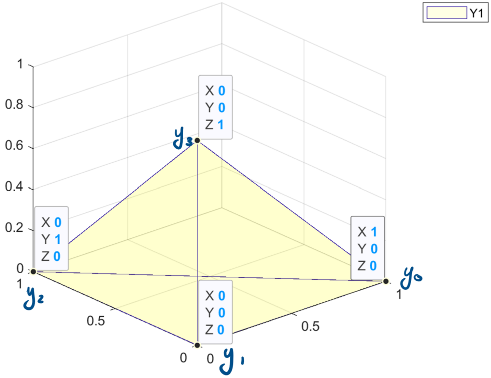
```
Y0a = [0 1 0 0;
       0 0 1 0;
       0 0 0 1];

del_e_s = 2;
del_oc_s = 1/2;
del_ic_s = -1/2;
gamma_s = 1/2;
```



1. Order

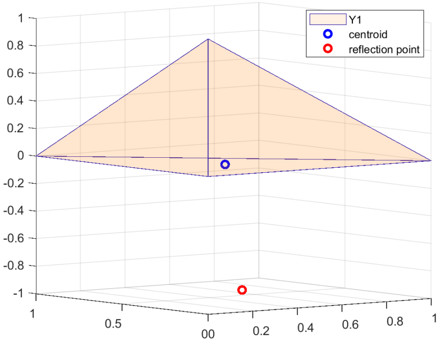
```
%ORDER THE initial simplex SIMPLEX
if iteration == 2
    for i = 1:k
        function_at_simplex(i) = f(current_Simplex(:,i));
        feval = feval +1;
    end
    feval_total(iteration-1) = feval;
    [current_Simplex,function_at_simplex] = sortSimplex(current_Simplex,function_at_simplex,stepComputed);
end
```

```
  case 'fullSort'
      %Do Insertion Sort to sort the simplex
      for i = 2:k
          key = Yi(:,i);
          fkey = fYi(i);
          j = i-1;
          while ((j>= 1) && (fkey < fYi(j)))
              Yi(:,j+1) = Yi(:,j);
              fYi(j+1) = fYi(j);
              j = j -1;
          end
          Yi(:,j+1) = key;
          fYi(j+1) = fkey;
      end
   case 'partialSort'
      %Sort the added vector Yn in the simplex
      k = length(Yi(1,:));
      for i = k:-1:2
          if fYi(i)< fYi(i-1)
              temp = Yi(:,i-1);
              Yi(:,i-1) = Yi(:,i);
              Yi(:,i) = temp;

              tempf = fYi(i-1);
              fYi(i-1) = fYi(i);
              fYi(i) = tempf;
          else
              break;
          end
      end
```



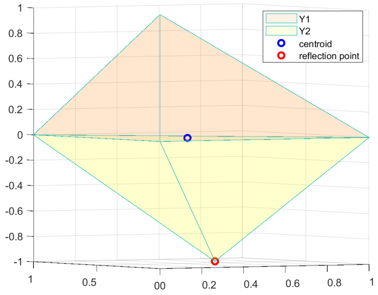
1 and 2. Calculate centroid and x^r

```
%CALCULATE CENTROID%
xc = zeros(k - 1,1);
for i = 1:k-1
    xc = xc + current_Simplex(:,i);
end
xc = (1/(k-1)).*xc;

%CALCULATE REFLECTION POINT%
xr = xc + (xc - current_Simplex(:,k));
fr = f(xr);
feval = feval + 1;
```



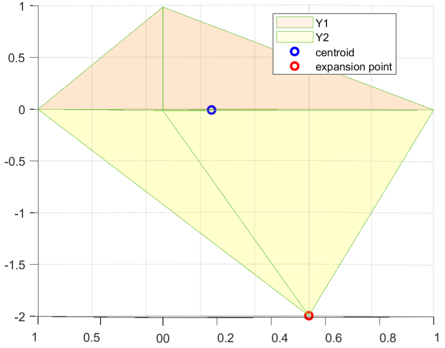
2. Reflect

```
%REFLECTION STEP%
if (function_at_simplex(1) <= fr)&&(fr < function_at_simplex(k-1))
    current_Simplex(:,k) = xr;
    function_at_simplex(k) = fr;
    stepComputed = "partialSort";
```



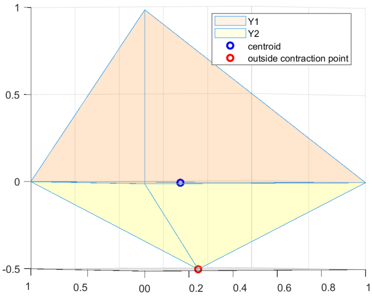
3. Expand

```
%EXPANSION%
elseif (fr < function_at_simplex(1))
    xe = xc + del_e*(xc - current_Simplex(:,k));
    fe = f(xe);
    feval = feval + 1;
    if fe < fr
        current_Simplex(:,k) = xe;
        function_at_simplex(k) = fe;
    else
        current_Simplex(:,k) = xr;
        function_at_simplex(k) = fr;
    end
    stepComputed = "partialSort";
```



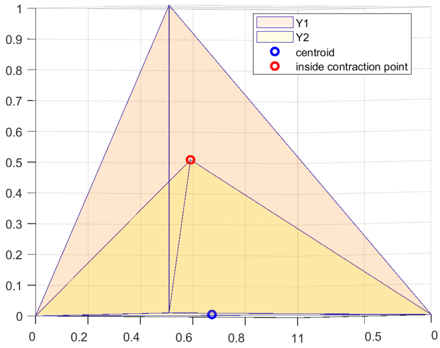
4.a) Outside Contraction

```
%OUTSIDE CONTRACTION%
elseif (function_at_simplex(k-1) <= fr)&&(fr < function_at_simplex(k))
    xoc = xc + del_oc*(xc - current_Simplex(:,k));
    foc = f(xoc);
    feval = feval + 1;
    stepComputed = "partialSort";
    if foc < fr
        current_Simplex(:,k) = xoc;
        function_at_simplex(k) = foc;
        stepComputed = "partialSort";
    else
        current_Simplex(:,k) = xr;
        function_at_simplex(k) = fr;
        stepComputed = "partialSort" ;
    end
```



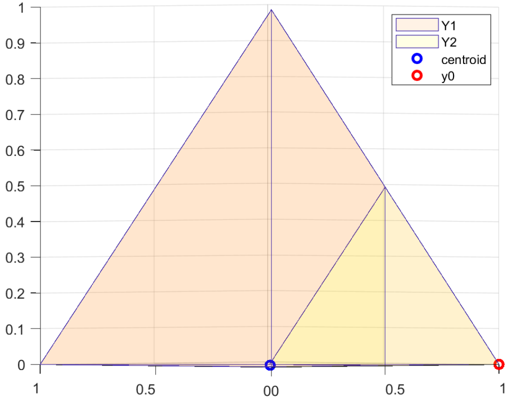
4.b) Inside Contraction

```
%INSIDE CONTRACTION%
elseif (fr >= function_at_simplex(k))
    xic = xc + del_ic.*(xc - current_Simplex(:,k));
    fic = f(xic);
    feval = feval + 1;
    if fic < function_at_simplex(k)
        current_Simplex(:,k) = xic;
        function_at_simplex(k) = fic;
        stepComputed = "partialSort";
```



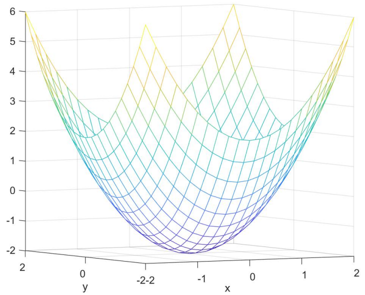
5. Shrink

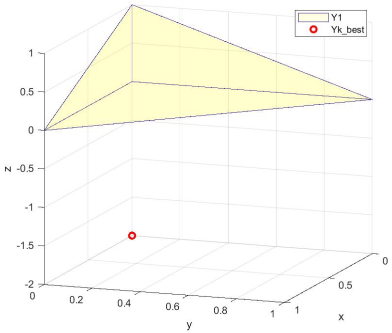
```
%SHRINK%
for i = 2:k
    current_Simplex(:,i) = (1-gamma).*current_Simplex(:,1) + gamma.*current_Simplex(:,i);
    function_at_simplex(i) = f(current_Simplex(:,i));
    feval = feval + 1;
end
stepComputed = "fullSort";
```



Examples of Nelder-Mead at work: A nice example

$$f(x) = x_1^2 + x_2^2 - 2$$



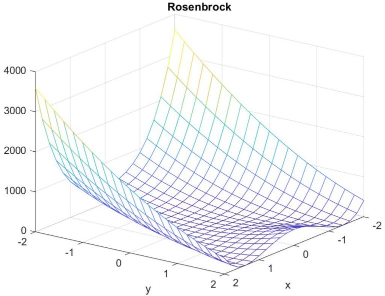


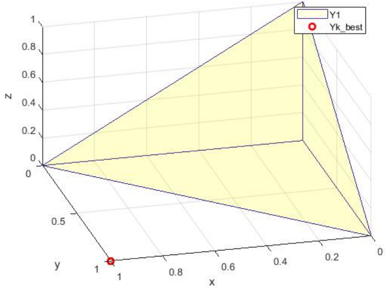
$$min(f(x)) = -2 \text{ for } x = (0,0)$$

$$f_{best}^k = -2 \text{ for } x_{best}^k = (0,0)$$

Examples of Nelder-Mead at work: Rosenbrock function

$f(x) = (1 - x_1)^2 + 100(x_2 - x_1)^2$



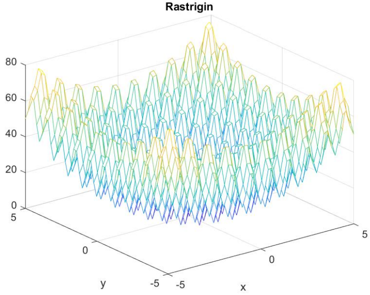


$min(f(x)) = 0$ for $x = (1, 1)$

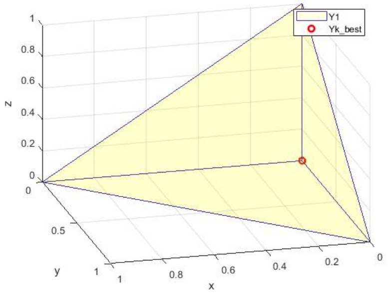
$f_{best}^{k} = 0$ for $x_{best}^{k} = (1, 1)$

Examples of Nelder-Mead at work: Rastrigin function

$f(x) =$
$20 + x_1^2 - 10cos(2\pi x_1) + x_2^2 - 10cos(2\pi x_2)$



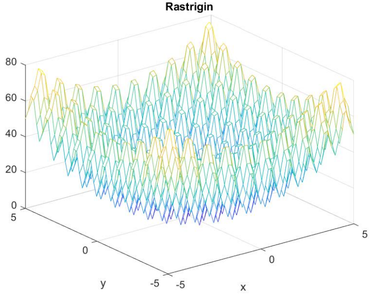
$min(f(x)) = 0$ for $x = (0, 0)$



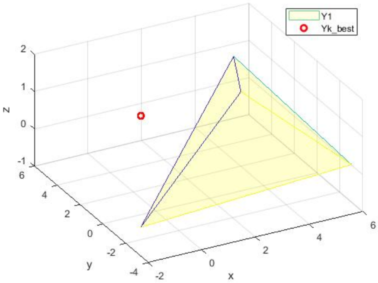
$f_{best}^k = 0$ for $x_{best}^k = (0, 0)$

Examples of Nelder-Mead at work: Rastrigin function

$f(x) =$
$20 + x_1^2 - 10cos(2\pi x_1) + x_2^2 - 10cos(2\pi x_2)$



$min(f(x)) = 0$ for $x = (0, 0)$



$f_{best}^k = 1.899$ for
$x_{best}^k = (-0.9950, -0.9950)$

Viscosity of a system

$$\eta(\dot{\gamma}) = \eta_0(1 + \lambda^2\dot{\gamma}^2)^{\frac{\beta-1}{2}}$$

A function of the strain rate, $\dot{\gamma}$, with parameters η_0, λ, and β.

TABLE 1.1. Observed data in rheology of polymeric systems

Observation i	Strain rate $\dot{\gamma}_i$ (s^{-1})	Viscosity η_i $(Pa \cdot s)$
1	0.0137	3220
2	0.0274	2190
3	0.0434	1640
4	0.0866	1050
5	0.137	766
6	0.274	490
7	0.434	348
8	0.866	223
9	1.37	163
10	2.74	104
11	4.34	76.7
12	5.46	68.1
13	6.88	58.2

Absolute Error:

$$\epsilon_i(\eta_0, \lambda, \beta) = \left|\eta_0(1 + \lambda^2\dot{\gamma}^2)^{\frac{\beta-1}{2}} - \eta_i\right|$$

Non-smooth optimization problem:

$$\hat{g}(\eta_0, \lambda, \beta) = \sum_{i=1}^{13} \epsilon_i(\eta_0, \lambda, \beta)$$

(Audet and Hare, 2017)

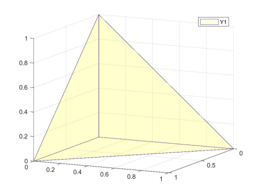
The Rheology Problem: Solved

1.a) Standard parameters with simplex A

Input:

```
Y0a = [0 1 0 0;
       0 0 1 0;
       0 0 0 1];
```

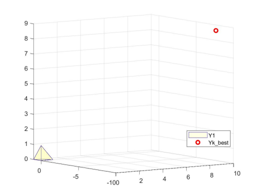
```
del_e_s = 2;
del_oc_s = 1/2;
del_ic_s = -1/2;
gamma_s = 1/2;
```



Output:

9.5062	9.5062	9.5062	9.5062
-8.4167	-8.4167	-8.4167	-8.4167
8.7269	8.7269	8.7269	8.7269

$$f^k_{best} = 32.7239$$



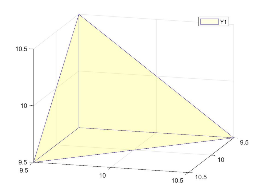
The Rheology Problem: Solved

1.b) Standard parameters with simplex B

Input:

```
Y0b = Y0a + (9.5)*ones(3,4);
```

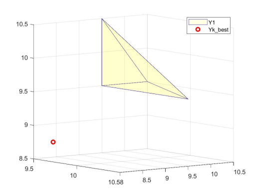
```
del_e_s = 2;
del_oc_s = 1/2;
del_ic_s = -1/2;
gamma_s = 1/2;
```



Output:

9.5062	9.5062	9.5062	9.5062
8.4167	8.4167	8.4167	8.4167
8.7269	8.7269	8.7269	8.7269

$$f_{best}^{k} = 32.7238$$



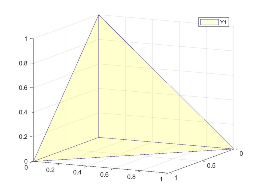
The Rheology Problem: Solved

2.a) New parameters with simplex A

Input:

```
Y0a = [0 1 0 0;
       0 0 1 0;
       0 0 0 1];
```

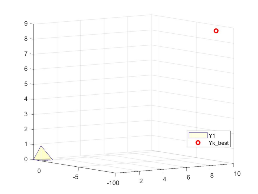
```
del_e_na = 4.71;
del_oc_na = 0.65;
del_ic_na = -0.3;
gamma_na = 0.94;
```



Output:

```
 9.5062    9.5062    9.5062    9.5062
-8.4167   -8.4167   -8.4167   -8.4167
 8.7269    8.7269    8.7269    8.7269
```

$$f_{best}^{k} = 32.7238$$

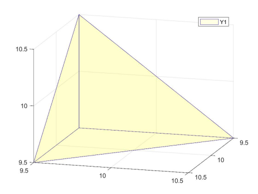


2.b) New parameters with simplex B

Input:

```
Y0b = Y0a + (9.5)*ones(3,4);
```

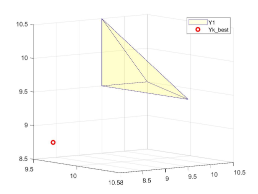
```
del_e_nb = 2.3;
del_oc_nb = 0.46;
del_ic_nb = -0.45;
gamma_nb = 0.01;
```



Output:

9.5062	9.5062	9.5062	9.5062
8.4167	8.4167	8.4167	8.4167
8.7269	8.7269	8.7269	8.7269

$$f_{best}^{k} = 32.7238$$



The Rheology Problem: Solved