Maths 3468 quizzes

4: Monday 26/3/12

(1) Show

$$\gamma_r + \gamma_s + \gamma_r\gamma_s \leq \gamma_{r+s}$$

when $r + s < 1/\epsilon_{\text{mach}}$.

Answer.

$$1 + \gamma_n = \frac{1}{1 - n\epsilon_{\text{mach}}}$$
$$1 + \gamma_r + \gamma_s + \gamma_r\gamma_s =$$
$$\frac{1}{1 - r\epsilon_{\text{mach}}}\frac{1}{1 - s\epsilon_{\text{mach}}} =$$
$$\frac{1}{1 - (r+s)\epsilon_{\text{mach}} + rs\epsilon_{\text{mach}}^2} \leq \frac{1}{1 - (r+s)\epsilon_{\text{mach}}} = 1 + \gamma_{r+s}$$

and subtract 1. ∎

(2) Show that solving $Ax = b$ by LU factorisation and (back) substitution (without pivoting), there is a perturbation ΔA such that

$$(A + \Delta A)\tilde{x} = b, \quad |\Delta A| \leq \gamma_{3n}|\tilde{L}||\tilde{U}|.$$

Answer. A perfect calculation would proceed as follows:

$$Ly = b$$
$$Ux = y$$

But L and U are rounded, and the calculations are also rounded. In place of

$$Ly = b$$

we have

$$\tilde{L}y = b$$

for which backward analysis of substitution yields

$$(\tilde{L} + \Delta_1\tilde{L})\tilde{y} = b$$

where $|\Delta_1\tilde{L}| \leq \gamma_n|\tilde{L}|$.

Then we solve

$$\tilde{U}x = \tilde{y}$$

and the rounded solution $\tilde{x}$ satisfies

$$(\tilde{U} + \Delta_2\tilde{U})\tilde{x} = \tilde{y}$$

where $|\Delta_2 \tilde{U}| \leq \gamma_n |\tilde{U}|$. Thus

$$(\tilde{L} + \Delta_1 \tilde{L})(\tilde{U} + \Delta_2 \tilde{U})\tilde{x} = b.$$

Write $\tilde{L}\tilde{U} = A + \Delta_3 A$, so

$$|\Delta_3 A| \leq \gamma_n \tilde{L}\tilde{U}|.$$

$$\begin{gathered}\tilde{L}\tilde{U} + \tilde{L}(\Delta_2\tilde{U}) + (\Delta_1\tilde{L})\tilde{U} + (\Delta_1\tilde{L})\Delta_2\tilde{U} = \\ A + \Delta_3 A + \tilde{L}(\Delta_2\tilde{U}) + (\Delta_1\tilde{L})\tilde{U} + (\Delta_1\tilde{L})\Delta_2\tilde{U} = \\ A + \Delta A\end{gathered}$$

where

$$|\Delta A| \leq (\gamma_n + \gamma_n + \gamma_n + \gamma_n^2)|\tilde{L}||\tilde{U}|$$

and the coefficient is

$$\leq \gamma_n + \gamma_{2n} \leq \gamma_{3n}.$$

(3) Maximise

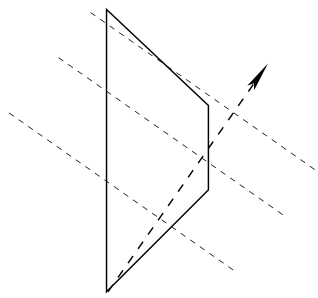
$$\begin{gathered}3x_1 + 2x_2 + 4x_3 \\ \text{subject to} \\ x_1 + x_2 + 2x_3 \leq 4 \\ 2x_1 \quad + 3x_3 \leq 5 \\ 2x_1 + x_2 + 3x_3 \leq 7 \\ x_1, x_2, x_3 \geq 0\end{gathered}$$

```
    1      1      2      1      0      0      4
    2      0      3      0      1      0      5
    2      1      3      0      0      1      7
    3      2      4      0      0      0      0

    0      0      0      4      5      7      0
                  ^in           ^
x_3 <= 4/2, 5/3, 7/3: 5/3       ^
                                ^out

   -1/3    1      0      1     -2/3    0      2/3
    2/3    0      1      0      1/3    0      5/3
    0      1      0      0     -1      1      2
    1/3    2      0      0     -4/3    0     -20/3

    0      0      5/3    2/3    0      2      20/3
           ^in           ^out
x_2 <= 2/3, x2 <= 2: x_4 out
```

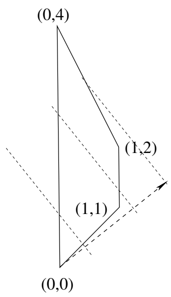
```
  -1/3   1     0     1    -2/3   0     2/3
   2/3   0     1     0     1/3   0     5/3
   1/3   0     0    -1    -1/3   1     4/3
   1     0     0    -2     0     0    -8

   0     2/3   5/3   0     0     4/3   8
   ^in
x_1<=5/2,4/1: x_1 = 5/2, x_3 out
                ^out


   0     1     1/2   1    -1/2   0     3/2
   1     0     3/2   0     1/2   0     5/2
   0     0    -1/2  -1    -1/2   1     1/2

   0     0    -3/2  -2    -1/2   0    -21/2

   5/2   3/2   0     0     0     1/2   21/2
```

The maximum, $21/2$, is found at $(5/2, 3/2, 0)$.

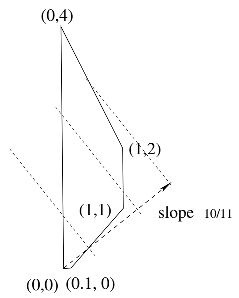
(4) The figure below indicates a feasible region (distorted square) and direction of maximally increasing cost which causes the simplex method, starting at O, to visit all corners. Construct such a linear programming problem explicitly and use the simplex method to solve it inefficiently.



Answer. Sorry, this is worse than I intended. The origin is on 3 edges, so this is degenerate, and the first step is ill-defined. Also, the direction of increasing cost needs to have a flatter slope and the top edge needs to have a steeper slope.



Let us begin with a nearby nondegenerate problem, by moving the bottom edge a bit to the right: $x_1 \le x_2 + 0.1$.




```
   1    -1    1    0    0    0.1   bottom edge
   1     0    0    1    0    1     second
   2     1    0    0    1    4     third

  11    10    0    0    0    0     cost
   0     0    0.1  1    4    0     initial point + 0 cost

x_1 in
x_1 <= 0.1;  x_1 <= 1; x_1 <= 2. x_3 goes out.

   1    -1    1    0    0    0.1
   0     1   -1    1    0    0.9
   0     3   -2    0    1    3.8

   0    21  -11    0    0   -1.1
   0.1   0    0  0.9  3.7    1.1

x_2 in
-x_2 <= 0.1; x_2 <= 0.9; x_2 <= 1.2333
x_2 in x_4 out


   1     0    0    1    0    1.0
   0     1   -1    1    0    0.9
   0     0    1   -3    1    1.1

   0     0   10  -21    0  -20

   1   0.9    0    0    1   20

x_3 in
0x_3 <= 1; -x_3 <= 0.9; x_3 <= 1.1,  So x_5 out.
```

```
   1      0     0     1     0     1.0
   0      1     0    -2     1     2.0
   0      0     1    -3     1     1.1

   0      0     0     9   -10   -31

   1      2     1     0   1.1    31

x_4 in.
x_4 <= 1; -2x_4 <= 1.9; -3x_4 <= 1.1; so x_4 <= 1 and x_1
goes out.

   1      0     0     1     0     1.0
   2      1     0     0     1     4.0
   3      0     1     0     1     4.1

  -9      0     0     0   -10   -40

   0      4     4.1   1     0    40
```

Returning to the degenerate problem. The first step is degenerate, but it changes the cost function.

```
   1     -1     1     0     0     0    bottom edge
   1      0     0     1     0     1    second
   2      1     0     0     1     4    third

  11     10     0     0     0     0    cost
   0      0     0     1     4     0    initial point + 0 cost

x_1 in
x_1 <= 0.1;  x_1 <= 1; x_1 <= 2. x_3 goes out.

   1     -1     1     0     0     0
   0      1    -1     1     0     1
   0      3    -2     0     1     4

   0     21   -11     0     0     0
   0      0     0     1     4     0

x_2 in
-x_2 <= 0.0; x_2 <= 1; x_2 <= 1.3333
x_2 in x_4 out
```

```
   1      0     0     1     0     1
   0      1    -1     1     0     1
   0      0     1    -3     1     1

   0      0    10   -21     0   -21

   1      1     0     0     1    21
```

```
x_3 in
0x_3 <= 1; -x_3 <= 1; x_3 <= 1,  So x_5 out.
```

```
   1      0     0     1     0     1
   0      1     0    -2     1     2
   0      0     1    -3     1     1

   0      0     0     9   -10   -31

   1      2     1     0     1    31
```

```
x_4 in.
x_4 <= 1; -2x_4 <= 2; -3x_4 <= 1; so x_4 <= 1 and x_1
goes out.
```

```
   1      0     0     1     0     1
   2      1     0     0     1     4
   3      0     1     0     1     4

  -9      0     0     0   -10   -40

   0      4     4     1     0    40
```

Looking at the points visited (with costs: the bottom lines in each step)

```
   0      0     0     1     4     0    initial point + 0 cost
   0      0     0     1     4     0
   1      1     0     0     1    21
   1      2     1     0     1    31
   0      4     4     1     0    40
```

the sequence is $(0, 0), (0, 0), (1, 1), (1, 2), (0, 4)$, so every corner is visited, although the optimum could have been reached in one step.