AMPL aid for the Thickeners project

1 Getting started

Once you hve the model files, you are set to go. The available files are
`thick.mod` (the model file)
`thick.dat` (the dat file)
We encourage you to look at the files to understand how the model is structured. You may now start AMPL by giving the command `ampl` in a terminal window. AMPL should start up and you get a prompt looking like

```
ampl:
```

2 AMPL

To load the model, type (all commands are terminated by ";")

```
ampl:  model thick.mod;
```

To load the data, type

```
ampl:  data thick.dat;
```

To obtain the optimal value, type

```
ampl:  solve;
```

You should get the result

```
MINOS 5.5:  optimal solution found.
8 iterations, objective -4217103.814
```

You may now take a closer look at the solution. To see the value of a variable, use the `display` command. As an example, to see the amount of Cellulose sent from the suppliers to the reactors, type
`ampl:  display y;` (see the `.mod`-file for the chosen variable names)
You should get the result

```
y :=
A R1 0
A R2 0
A R3 0
B R1 3375.57
B R2 878.986
B R3 2444.44
C R1 0
C R2 0
C R3 0
D R1 0
D R2 0
D R3 0
```

To obtain the reduced costs for a variable, type

```
ampl:  display y.rc;
```

In the same fashion, you may get the dual variables corresponding to the constraints by typing

```
ampl:  display Reac_Cap_Cell.dual;
```

and the slack in the the constraints by typing

```
ampl:  display Reac_Cap_Cell.slack;
```

If you want to aggregate values you may use summation in the displayed expressions. For example, to get the total amount of Cellulose bought, you may type

```
ampl:  display sum{i in I, j in J} x[i,j];
```

If you want to get specific elements you can index the variables and constraints, for example

```
ampl:  display x['A','R3'];
```

returns the amount of Cellulose bought from supplier A to be used in reactor R3 (Gamlestaden).

If you change the model and/or the data and wish to reload them, before loading the `.mod` and/or the `.dat`-file once again you must type either

```
ampl:  reset;
```

to reset everything, or

```
ampl:  reset data;
```

to reset everything from the `.dat`-file. If you do not do this AMPL will complain as it will believe that you are redefining variables and parameters.

Constants may be changed by using the `let` command. As an example, the command

```
ampl:  let rcapcell['R1'] := 1000;
```

will set a new value on the cellulose capacity in reactor R1 (Mölndal).

2.1 Using scripts

Sometimes it might be useful do write scripts that execute the AMPL commands. For example for parametric studies or other iterative processes it can be used to reduce the work load. The script is written in a file with suffix `.run` and then called directly from the terminal for example by typing

```
> ampl thick.run
```

A simple `.run`-file can look like

```
reset;
model thick.mod;
data thick.dat;
solve;
display x;
display Reac_Cap_Cell.dual;
```

It is also possible to use loops and to save values in text files. To learn more about AMPL, see `www.ampl.com`.

3 The most probable mistakes

Q: I typed a command, but nothing happened. When I type the next command I get weird errors such as

```
syntax error
context:  >>>...............
```

A: You probably forgot a ";" after your last command. If nothing happens, look at the prompt. If it reads

```
ampl ?
```

then AMPL is expecting the rest of the last command (just type ";" if you forgot it before).

Q: I get errors of the type

```
invalid subscript my_variable[j,i]
```

although it has indices i and j.
A: Check the order of your indices.

Q: I get errors of the type

```
th.mod, line 5 (offset 55):
     I is already defined
context:  set >>> I; <<<
```

A: Do not forget to reset the model before loading a new one.