SOLVING SYSTEMS OF LINEAR EQUATIONS WITH GAMS

ERWIN KALVELAGEN

Abstract. This document describes some issues with respect to solving systems of linear equations $Ax = b$ using GAMS.

1. Introduction

Solving a system of linear equations $Ax = b$, where A is a (square) matrix, is a task GAMS and LP solvers are well equipped to do. As LP solvers do not require A to be square, even more general systems can be solved. Even if the matrix A is singular, an LP solver will report a solution, provided there exists a solution.

The trick that needs to be used to shoe horn such a model into a linear optimization model is to select a variable to optimize. One can just pick a variable that appears in the model, especially when it is known that there exists just one unique solution. However, from a modeling point of view it may be better to introduce a *dummy* variable. This makes clear that this variable has no intrinsic meaning in the model, but just is created for syntactical reasons. Below is a small example of a system of linear equations that is not square. We imposed bounds on p_i, which helps to detect modeling errors: the model will be declared infeasible if $p_i < 0$ or $p_i > 1$.

Model markov.gms. [1]

```
$ontext

   Given a Markov transition matrix, find the steady state
   probabilities p(i).

   Erwin Kalvelagen, 2001

$offtext


set i 'states' /state1*state4/;

alias (i,j);

table A(i,j) 'transition matrix'
         state1   state2   state3   state4
state1    0.1      0.2      0.3      0.4
state2    0.9
state3             0.8
state4                      0.7      0.6
;


variables
   dummy  'dummy objective variable'
   p(i)   'steady state probabilities'
;
```

Date: February 2008.

[1] http://amsterdamoptimization.com/models/lineq/markov.gms

```
positive variables p;
p.up(i) = 1;


equations
   steadystate(i)
   normalize
   edummy
;

steadystate(i)..  sum(j, A(i,j)*p(j)) =e= p(i);
normalize..       sum(i, p(i)) =e= 1;
edummy..          dummy =e= 0;

model m1 /steadystate,normalize,edummy/;
solve m1 minimizing dummy using lp;
```

Note that if we have a square model $Ax = b$ with a nonsingular matrix we can solve the system of equations very efficiently: namely in zero iterations. We know that all variables are basic and all equations are non-basic in the final solution. Using an advanced basis, we can tell the solver to use such an optimal basis.

Model zero.gms. [2]

```
$ontext

Solve

  3x  +  2y     + 4z  =   5
  4x  -  3y    - 7z  =  17
  2x  -  3y    + 6z  =   2

in zero iterations.

$offtext

variables x,y,z,dummy;

equations e1,e2,e3,edummy;

e1..  3*x  +  2*y    + 4*z  =e=   5;
e2..  4*x  -  3*y    - 7*z  =e=  17;
e3..  2*x  -  3*y    + 6*z  =e=   2;
edummy.. dummy =e= 0;

*
* the variables are basic
*
x.m=0;
y.m=0;
z.m=0;
dummy.m=0;


*
* the equations are nonbasic
*
e1.m = 1;
e2.m = 1;
e3.m = 1;
edummy.m = 1;

model m /all/;
solve m using lp minimizing dummy;
```

Indeed this model solves in zero iterations:

[2] http://amsterdamoptimization.com/models/lineq/zero.gms

```
               S O L V E      S U M M A R Y

     MODEL   m                   OBJECTIVE  dummy
     TYPE    LP                  DIRECTION  MINIMIZE
     SOLVER  BDMLP               FROM LINE  40

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL
**** OBJECTIVE VALUE                0.0000

 RESOURCE USAGE, LIMIT          0.000      1000.000
 ITERATION COUNT, LIMIT         0         10000
```

If the matrix A is not square, or if A is singular, the above approach can still be used, but the algorithm will need a few iterations to move some slack variables into the basis.

2. Calculating the inverse

To find the inverse of a matrix, we can use (or misuse) the looping facilities in GAMS to implement a Gaussian elimination procedure. An example of a GAMS model that implements this strategy can be found in the GAMS model library: `gauss.gms`[3].

A different approach is to find the inverse column by column. We can solve:

$$Ax_i = e_i \tag{1}$$

where e_i is the i^{th} column of the identity. Then the columns $(x_1, .., x_n)$ form the inverse matrix.

A different approach is to write down

$$AX = I \tag{2}$$

as a set of linear equations that need to be solved for $X = A^{-1}$.

A GAMS model that implements this, can be written as:

Model inverse.gms. [4]

```
$ontext

   Finds the inverse of a matrix

   Erwin Kalvelagen, nov 2002

   Reference: model gauss.gms from the model library
              http://www.gams.com/modlib/libhtml/gauss.htm

$offtext

set i  /i1*i3 /;
alias (i,j,k);

table a(i,j) 'original matrix'
      i1     i2     i3
i1    1      2      3
i2    1      3      4
i3    1      4      3
;

parameter ident(i,j) 'identity matrix';
ident(i,i) = 1;
```

[3] http://www.gams.com/modlib/libhtml/gauss.htm

[4] http://amsterdamoptimization.com/models/lineq/inverse.gms

```
*
* method 1: solve a series of models Ax=b where
* b is a column from the unit matrix.
*


variable
   dummy   'dummy objective variable'
   col(j)  'column of inv(A)'
;

equations
   edummy   'dummy objective function'
   lineq(i) 'system of linear equations'
;

parameter b(i) 'rhs';

edummy..    dummy =e= 0;
lineq(i)..  sum(k, a(i,k)*col(k)) =e= b(i);

model m1 /lineq,edummy/;

parameter inverse(i,j) 'inv(A)';

loop(j,
    b(i) = ident(i,j);
    solve m1 minimizing dummy using lp;
    inverse(i,j) = col.l(i);
);

display inverse;

*
* method 2: solve a bigger system AX=I
*

variable ainv(i,j) 'inverse of A';

equation lineqall(i,j);
lineqall(i,j)..  sum(k, a(i,k)*ainv(k,j)) =e= ident(i,j);

model m2 /lineqall,edummy/;
solve m2 minimizing dummy using lp;

display ainv.l;
```

The resulting solution for

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 4 \\ 1 & 4 & 3 \end{pmatrix} \tag{3}$$

is given by:

```
----     42 VARIABLE ainv.L  inverse of A

            i1          i2          i3

i1       3.500      -3.000       0.500
i2      -0.500                   0.500
i3      -0.500       1.000      -0.500
```

3. Systems of complex linear equations

3.1. **Solving $Ax = b$, the complex case.** GAMS and all the solvers underneath GAMS are not equipped to deal with complex numbers. Complex numbers have the form $z = a + ib$ with i defined by the famous expression $i^2 = -1$. a is called the real part of z and b forms the imaginary part. We can map a complex system of equations in $\mathcal{C}^n$ onto a problem in $\mathcal{R}^{2n}$. This allows us to solve the problem using standard tools.

A system of equations with complex variables and coefficients $Ax = b$ can be expanded into:

$$\begin{pmatrix} re(A) & -im(A) \\ im(A) & re(A) \end{pmatrix} \begin{pmatrix} re(x) \\ im(x) \end{pmatrix} = \begin{pmatrix} re(b) \\ im(b) \end{pmatrix} \tag{4}$$

As an example consider the problem:

$$\begin{aligned} (1+i)z + (2-i)w &= 2 + 7i \\ 7z + (8-2i)w &= 4 - 9i \end{aligned} \tag{5}$$

The following model shows a simple implementation:

Model complex.gms. [5]

```
$ontext

   Solution of a system of complex equations

   Erwin Kalvelagen, April 2002

$offtext


set i /i1*i2/;
alias (i,j);

table data(*,i,*)
          i1   i2   rhs
real.i1    1    2    2
real.i2    7    8    4

imag.i1    1   -1    7
imag.i2       -2   -9
;

variables
   rx(i) 'real part'
   ix(i) 'imaginary part'
   dummy 'dummy objective variable'
;

equations
   real(i)  'real part'
   imag(i)  'imaginary part'
   dummyobj 'dummy objective'
;

dummyobj.. dummy=e=0;

real(i)..
   sum(j, data('real',i,j)*rx(j)-data('imag',i,j)*ix(j)) =e= data('real',i,'rhs');

imag(i)..
   sum(j, data('imag',i,j)*rx(j)+data('real',i,j)*ix(j)) =e= data('imag',i,'rhs');
```

[5] http://amsterdamoptimization.com/models/lineq/complex.gms

```
model m /all/;

solve m using lp minimizing dummy;

display rx.l, ix.l;
```

The solution of this system is:

$$z = \frac{838}{185} - \frac{699}{185}i \qquad (6)$$
$$w = -\frac{698}{185} + \frac{229}{185}i$$

The solution of the GAMS model is identical to this solution:

```
----     47 VARIABLE  rx.L  real part

i1  4.530,    i2 -3.773


----     47 VARIABLE  ix.L  imaginary part

i1 -3.778,    i2  1.238
```

3.2. **Inverting a complex matrix.** The same approach can be used to find the (complex) inverse of a complex matrix. We search A^{-1} such that:

$$\begin{pmatrix} re(A) & -im(A) \\ im(A) & re(A) \end{pmatrix} \begin{pmatrix} re(A^{-1}) \\ im(A^{-1}) \end{pmatrix} = \begin{pmatrix} I \\ 0 \end{pmatrix} \qquad (7)$$

which can be written out as:

$$\sum_k re(a_{i,k})re(a^{-1}_{k,j}) - im(a_{i,k})im(a^{-1}_{k,j}) = e_{i,j} \qquad (8)$$
$$\sum_k im(a_{i,k})re(a^{-1}_{k,j}) + re(a_{i,k})im(a^{-1}_{k,j}) = 0$$

The model below calculates the inverse matrix of

$$\begin{pmatrix} 1+i & 2-i \\ 7 & 8-2i \end{pmatrix} \qquad (9)$$

Model complexinv.gms. [6]

```
$ontext

Using the example from http://amsterdamoptimization.com/pdf/lineq.pdf,
the code below first calculates the complex inverse ainv,
and then calculates x = inv(a)*b.

Erwin Kalvelagen, May 2004

$offtext

set i /i1*i2/;
alias (i,j,k);

table data(*,i,*)
          i1   i2   rhs
real.i1    1    2    2
real.i2    7    8    4

imag.i1    1   -1    7
imag.i2        -2   -9
;
```

[6] http://amsterdamoptimization.com/models/lineq/complexinv.gms

```
set cmplx /re,im/;
parameter a(i,j,cmplx);
a(i,j,'re') = data('real',i,j);
a(i,j,'im') = data('imag',i,j);

parameter ident(i,j) 'identity matrix';
ident(i,i) = 1;

parameter b(i,cmplx);
b(i,'re') = data('real',i,'rhs');
b(i,'im') = data('imag',i,'rhs');

display a,ident;

variables
   ainv(i,j,cmplx) 'inverse matrix'
   dummy 'dummy objective variable'
;

equations
   real(i,j)
   imag(i,j)
   dummyobj 'dummy objective'
;

dummyobj.. dummy=e=0;

real(i,j)..
   sum(k, a(i,k,'re')*ainv(k,j,'re')-a(i,k,'im')*ainv(k,j,'im')) =e= ident(i,j);
imag(i,j)..
   sum(k, a(i,k,'im')*ainv(k,j,'re')+a(i,k,'re')*ainv(k,j,'im')) =e= 0;

model m /all/;

solve m using lp minimizing dummy;

display ainv.l;

parameter x(i,cmplx);
x(i,'re') = sum(j, ainv.l(i,j,'re')*b(j,'re') - ainv.l(i,j,'im')*b(j,'im'));
x(i,'im') = sum(j, ainv.l(i,j,'re')*b(j,'im') + ainv.l(i,j,'im')*b(j,'re'));
display b,x;
```

4. Laplace's equation

4.1. **Solving a PDE.** Discretization of Partial Differential Equations (PDE's) using finite difference approximation scheme's leads to large systems of linear equation. Consider the numerical solution of the Laplace's Equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{10}$$

on a square, where the boundary values of u are given. Assuming a grid of the form:

$$\begin{array}{ccc} & u_{i,j+1} & \\ u_{i-1,j} & u_{i,j} & u_{i+1,j} \\ & u_{i,j-1} & \end{array} \tag{11}$$

then the first derivatives can be approximated by:

$$\begin{aligned} \frac{\partial u}{\partial x} &\approx \frac{u_{i+1,j} - u_{i,j}}{h} \\ \frac{\partial u}{\partial y} &\approx \frac{u_{i,j+1} - u_{i,j}}{h} \end{aligned} \tag{12}$$

and the second derivatives by:

$$\begin{aligned}\frac{\partial^2 u}{\partial x^2} &\approx \frac{1}{h}\left(\frac{u_{i+1,j}-u_{i,j}}{h}-\frac{u_{i,j}-u_{i-1,j}}{h}\right)\\ &= \frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{h^2}\\ \frac{\partial^2 u}{\partial y^2} &\approx \frac{1}{h}\left(\frac{u_{i,j+1}-u_{i,j}}{h}-\frac{u_{i,j}-u_{i,j-1}}{h}\right)\\ &= \frac{u_{i,j+1}-2u_{i,j}+u_{i,j-1}}{h^2}\end{aligned} \tag{13}$$

Combining this we can approximate

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0 \tag{14}$$

by

$$\frac{1}{h^2}(u_{i,j+1}+u_{i,j-1}+u_{i-1,j}+u_{i+1,j}-4u_{i,j})=0 \tag{15}$$

or

$$4u_{i,j}=u_{i,j+1}+u_{i,j-1}+u_{i-1,j}+u_{i+1,j} \tag{16}$$

As an example consider the problem of solving the steady-state heat conduction problem over the unit square. This can be formulated as solving Laplace's Equation subject to the Dirichlet boundary conditions:

$$\begin{aligned}u(x,0)&=300\\ u(x,1)&=u(0,y)=u(1,y)=0\end{aligned} \tag{17}$$

This model can be implemented in GAMS as follows:

Model laplace.gms. [7]

```
$ontext

   Finite difference method of Laplace's Equation

          Uxx + Uyy = 0

   subject to the Dirichlet boundary condition

   Erwin Kalvelagen, april 2003

$offtext


scalar b 'boundary value for y=0' /300/;

set i 'interior nodes'  /i1*i99/;
alias (i,j);

set j0(j);
j0(j)$(ord(j)=1) = yes;

variables u(i,j),dummy;
equation e(i,j),obj;

obj.. dummy =e= 0;
e(i,j)..  4*u(i,j) =e=  u(i,j+1) + u(i,j-1) + u(i-1,j) + u(i+1,j) + b$(j0(j));

model m /all/;
```

[7] http://amsterdamoptimization.com/models/lineq/laplace.gms

```
solve m using lp minimizing dummy;

display u.l;
```

4.2. **Advanced basis.** This is quite a difficult LP, and many solvers will encounter numerical difficulties. The solution is again to provide a good initial basis:

```
obj.m=1;
e.m(i,j)=1;
dummy.m=0;
u.m(i,j)=0;
```

This will significantly speed up the solution times. E.g. for a few solvers we have the following results:

Solver	**Default**			**Advanced basis**		
	its	secs	note	its	secs	note
BDMLP			failure			failure
MINOS			unbounded	0	0.5	
CPLEX	4358	59.843		0	8	
XPRESS	9801	55.093		0	1.109	
MOSEK	0	54.812		0	57.656	

Some solvers really appreciate the advanced basis resulting in a significant improvement in performance.

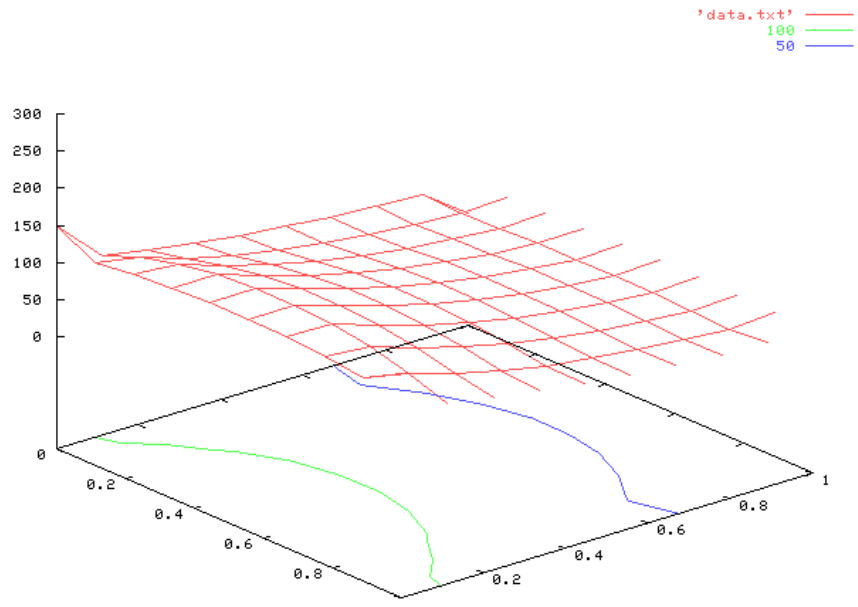

FIGURE 1. GNUPLOT 3D plot of the results of `laplace.gms`

4.3. **Plotting the solution.** We can export the solution to a GNUPLOT data file using:

```
parameter v(i);
v(i) = ord(i)/(1+card(i));
file f /data.txt/;
put f;
loop((i,j),
  put v(i), v(j), u.l(i,j)/;
);
```

The result is shown in figure 1.

5. A large input-output matrix

5.1. **Introductory theory.** Input-output analysis deals with the inter-relationships between sectors of an economy. Sectors use intermediate products from other products and deliver goods to other sectors. Monetary flows go the other way around.

The I-O table records the transactions between sectors. Often they are organized on yearly data, and several aggregation levels. A small I-O table deals with a handful of sectors while a highly disaggregated I-O table may record the transactions of hundreds of sectors.

The basic structure of an input-output transaction table is depicted in table 5.1.

	Intermediate Demand	Final Demand	Domestic Production
Intermediate Inputs	$x_{1,1} \cdots x_{1,n}$ $\vdots \quad \vdots$ $x_{n,1} \cdots x_{n,n}$	Y_1 $\vdots$ Y_n	X_1 $\vdots$ X_n
Value Added	$V_1 \cdots V_n$		
Domestic Production	$X_1 \cdots X_n$		

Table 1. Transaction table structure

The entries $x_{i,j}$ can be interpreted as the output of sector i that is used by sector j as input. The following relations hold:

$$\sum_j x_{i,j} + Y_i = X_i \tag{18}$$

and

$$\sum_i x_{i,j} + V_j = X_j \tag{19}$$

The matrix of input-coefficients A is defined by

$$a_{i,j} = x_{i,j}/X_j \tag{20}$$

This leads to the equation

$$AX + Y = X \tag{21}$$

or

$$(I - A)X = Y \tag{22}$$

which gives:

$$X = (I - A)^{-1}Y \tag{23}$$

The matrix $(I - A)^{-1}$ is often called the *Leontief Inverse Matrix*. Calculating this inverse can be accomplished in GAMS by defining a system of linear equations

$$(I - A)^{-1}I = (I - A) \tag{24}$$

A different way is to use the expansion:

$$(I - A)^{-1} = I + A + A^2 + A^3 \cdots \tag{25}$$

We can easily derive this series expansion by writing:

$$\begin{aligned} &(I - A)(I + A + A^2 + A^3 \cdots) = \\ &(I + A + A^2 + A^3 \cdots) - (A + A^2 + A^3 \cdots) = \\ &I \end{aligned} \tag{26}$$

if $\lim_{k\to\infty} A^k = 0$.

This series converges as we have elements with a range $(-1, 1)$. In fact for the large example below, the maximum element of A^k is as follows:

```
      441 PARAMETER maxelem             =        0.214
      441 PARAMETER maxelem             =        0.099
      441 PARAMETER maxelem             =        0.046
      441 PARAMETER maxelem             =        0.022
      441 PARAMETER maxelem             =        0.010
      441 PARAMETER maxelem             =        0.005
      441 PARAMETER maxelem             =        0.002
      441 PARAMETER maxelem             =        0.001
      441 PARAMETER maxelem             =  5.984566E-4
      441 PARAMETER maxelem             =  2.992630E-4
```

i.e. the series converges quickly.

The matrix $(I - A)^{-1}$ is convenient to evaluate scenario's

$$\Delta X = (I - A)^{-1}\Delta Y \tag{27}$$

5.2. **A real-world example: Input-Output table for Japan, 1995.** To illustrate the above we downloaded a real-world 93 sector I-O table from Japan for 1995 [3]. The data came in form of an Excel spreadsheet, so we used **XLS2GMS** to convert this to a GAMS include file. The include file is too large to completely reproduce here, but a fragment is listed below:

```
* ------------------------------------------------------
* XLS2GMS Version 1.4, December 2001
* Erwin Kalvelagen, GAMS Development Corp.
* ------------------------------------------------------
* Application: Microsoft Excel
* Version:     9.0
* Workbook:    D:\gams projects\book\io93pro.xls
* Sheet:       Producers' Prices
* Range:       $A$2:$DJ$106
* ------------------------------------------------------
    001     002     003     004     005     006     007     008     009     010      011 ....
001 177243  378270  6692    916     0       0       0       0       0       3945433  562475
002 64674   257455  9751    3063    0       0       0       0       0       2115243  0
003 375363  172254  0       0       0       0       0       0       0       0        0
004 1694    0       0       344226  0       124     109     587     18      14439    0
005 0       0       0       0       130498  0       0       0       0       1421408  0
006 0       0       0       0       0       0       0       0       0       0        0
007 0       0       0       0       0       0       4081    0       0       0        0
008 0       0       0       0       0       0       0       27      0       0        0
009 0       0       0       0       0       0       0       0       25      0        0
010 0       4886    0       24305   94167   0       0       0       0       4193486  409067
```

```
011 0       2473    0      0       12377   0     0       0      0     42914    140319
 :
 :
```

The GAMS model to read this table and do some elementary checks is reproduced below. It shows that GAMS has no problem in dealing with larger data sets, and that the algebra is not more complicated than for small problems.

leontief.gms. [8]

```
$ontext

       93 sector input/output table (japan, 1995).

       Erwin Kalvelagen, may 2002


       Data from: http://www.stat.go.jp/english/data/io/

$offtext



set ia 'all sectors (input), including subtotals' /
   001  'Crop cultivation'
   002  'Livestock and sericulture'
   003  'Agricultural services'
   004  'Forestry'
   005  'Fisheries'
   006  'Metallic ores'
   007  'Non-metallic ores'
   008  'Coal'
   009  'Crude petroleum and natural gas'
   010  'Foods'
   011  'Drinks'
   012  'Feeds and organic fertilizer, n.e.c.'
   013  'Tabacco'
   014  'Textile products'
   015  'Wearing apparel and other textile products'
   016  'Timber and wooden products'
   017  'Furniture and fixtures'
   018  'Pulp, paper, paperboard and processed paper'
   019  'Paper products'
   020  'Publishing and printing'
   021  'Chemical fertilizer'
   022  'Inorganic basic chemical products'
   023  'Petrochemical basic products and intermediate chemical products'
   024  'Synthetic resins'
   025  'Synthetic fibers'
   026  'Medicaments'
   027  'Final chemical products, n.e.c.'
   028  'Petroleum refinery products'
   029  'Coal products'
   030  'Plastic products'
   031  'Rubber products'
   032  'Leather, fur skins and miscellaneous leather products'
   033  'Glass and glass products'
   034  'Cement and cement products'
   035  'Pottery, china and earthenware'
   036  'Other ceramic, stone and clay products'
   037  'Pig iron and crude steel'
   038  'Steel products'
   039  'Steel castings and forgings and other steel products'
   040  'Non-ferrous metals'
   041  'Non-ferrous metal products'
   042  'Metal products for construction and architecture'
   043  'Other metal products'
```

[8] http://amsterdamoptimization.com/models/lineq/leontief.gms and http://amsterdamoptimization.com/models/lineq/io93pro.inc

```
   044  'General industrial machinery'
   045  'Special industrial machinery'
   046  'Other general machines'
   047  'Machinery for office and service industry'
   048  'Household electric appliance'
   049  'Electronic equipment and communication equipment'
   050  'Heavy electrical equipment'
   051  'Other electrical machinery'
   052  'Motor vehicles'
   053  'Ships and repair of ships'
   054  'Other transportation equipment and repair of transportation equipment'
   055  'Precision instruments'
   056  'Miscellaneous manufacturing products'
   057  'Construction'
   058  'Repair of constructions'
   059  'Civil engineering'
   060  'Electric power for enterprise use'
   061  'Gas and heat supply'
   062  'Water supply'
   063  'Waste disposal services'
   064  'Commerce'
   065  'Finance and insurance'
   066  'Real estate agencies and rental services'
   067  'House rent'
   068  'Railway transport'
   069  'Road transport(except transport by private cars)'
   070  'Transport by private cars'
   071  'Water transport'
   072  'Air transport'
   073  'Freight forwarding'
   074  'Storage facility services'
   075  'Services relating to transport'
   076  'Communication'
   077  'Broadcasting'
   078  'Public administration'
   079  'Education'
   080  'Research'
   081  'Medical service and health'
   082  'Social security'
   083  'Other public services'
   084  'Advertising, survey and information services'
   085  'Goods rental and leasing services'
   086  'Repair of motor vehicles and machine'
   087  'Other business services'
   088  'Amusement and recreational services'
   089  'Eating and drinking places'
   090  'Hotel and other lodging places'
   091  'Other personal services'
   092  'Office supplies'
   093  'Activities not elsewhere classified'
   094  'Total of intermediate sectors'
   096  'Consumption expenditures outside households(column)'
   097  'Compensation of employees'
   098  'Operating surplus'
   099  'Depreciation of fixed capital'
   100  'Indirect taxes(excluding custom duties and commodity taxes on imported goods)'
   101  '(less)Current subsidies'
   112  'Total of gross value added sectors'
   115  'Total domestic products(gross inputs)'
   116  'Net domestic product(Factor cost)'
   117  'Gross domestic products'
/;

set ja 'all sectors (output), including subtotals' /

   001  'Crop cultivation'
   002  'Livestock and sericulture'
   003  'Agricultural services'
   004  'Forestry'
   005  'Fisheries'
   006  'Metallic ores'
   007  'Non-metallic ores'
```

```
008  'Coal'
009  'Crude petroleum and natural gas'
010  'Foods'
011  'Drinks'
012  'Feeds and organic fertilizer, n.e.c.'
013  'Tabacco'
014  'Textile products'
015  'Wearing apparel and other textile products'
016  'Timber and wooden products'
017  'Furniture and fixtures'
018  'Pulp, paper, paperboard and processed paper'
019  'Paper products'
020  'Publishing and printing'
021  'Chemical fertilizer'
022  'Inorganic basic chemical products'
023  'Petrochemical basic products and intermediate chemical products'
024  'Synthetic resins'
025  'Synthetic fibers'
026  'Medicaments'
027  'Final chemical products, n.e.c.'
028  'Petroleum refinery products'
029  'Coal products'
030  'Plastic products'
031  'Rubber products'
032  'Leather, fur skins and miscellaneous leather products'
033  'Glass and glass products'
034  'Cement and cement products'
035  'Pottery, china and earthenware'
036  'Other ceramic, stone and clay products'
037  'Pig iron and crude steel'
038  'Steel products'
039  'Steel castings and forgings and other steel products'
040  'Non-ferrous metals'
041  'Non-ferrous metal products'
042  'Metal products for construction and architecture'
043  'Other metal products'
044  'General industrial machinery'
045  'Special industrial machinery'
046  'Other general machines'
047  'Machinery for office and service industry'
048  'Household electric appliance'
049  'Electronic equipment and communication equipment'
050  'Heavy electrical equipment'
051  'Other electrical machinery'
052  'Motor vehicles'
053  'Ships and repair of ships'
054  'Other transportation equipment and repair of transportation equipment'
055  'Precision instruments'
056  'Miscellaneous manufacturing products'
057  'Construction'
058  'Repair of constructions'
059  'Civil engineering'
060  'Electric power for enterprise use'
061  'Gas and heat supply'
062  'Water supply'
063  'Waste disposal services'
064  'Commerce'
065  'Finance and insurance'
066  'Real estate agencies and rental services'
067  'House rent'
068  'Railway transport'
069  'Road transport(except transport by private cars)'
070  'Transport by private cars'
071  'Water transport'
072  'Air transport'
073  'Freight forwarding'
074  'Storage facility services'
075  'Services relating to transport'
076  'Communication'
077  'Broadcasting'
078  'Public administration'
079  'Education'
```

```
   080  'Research'
   081  'Medical service and health'
   082  'Social security'
   083  'Other public services'
   084  'Advertising, survey and information services'
   085  'Goods rental and leasing services'
   086  'Repair of motor vehicles and machine'
   087  'Other business services'
   088  'Amusement and recreational services'
   089  'Eating and drinking places'
   090  'Hotel and other lodging places'
   091  'Other personal services'
   092  'Office supplies'
   093  'Activities not elsewhere classified'
   094  'Total of intermediate sectors'
   096  'Consumption expenditure outside households(row)'
   097  'Consumption expenditure(private)'
   098  'Consumption expenditure of general government'
   099  'Gross domestic fixed capital formation(public)'
   100  'Gross domestic fixed capital formation(private)'
   101  'Increase in stocks'
   102  'Total domestic final demand'
   103  'Total domestic demand'
   104  'Exports'
   105  'Balancing sector'
   106  'Total final demand'
   107  'Total demand'
   108  '(less)Imports'
   109  '(less)Custom duties'
   110  '(less)Commodity taxes on imported goods'
   111  '(less)Total imports'
   112  'Total of final demand sectors'
   115  'Total domestic products(gross outputs)'
   117  'Gross National expenditure'

/;

table t(ia,ja) 'transaction table -- producers prices in million yens'
$include io93pro.inc
;

set i(ia) 'intermediate inputs' / 001*093 /;
set j(ja) 'intermediate demand' / 001*093 /;


*
* consistency checks
*
scalar maxres  'max residual';
maxres = smax(i, abs( sum(j, t(i,j)) + t(i,'102') - t(i,'103')));
display "row balance",maxres;
maxres = smax(j, abs( sum(i, t(i,j)) + t('112',j) - t('115',j)));
display "column balance",maxres;


parameter a(i,j) 'input coefficient table';
a(i,j) = t(i,j)/t('115',j);


*---------------------------------------------------------------
* calculate the Leontief Inverse matrix by solving a system
* of linear equations.
* This is an almost completely dense problem so not too
* easy. BDMLP needs more memory than the default to solve this.
*---------------------------------------------------------------


variables
   inva(i,j) 'Leontief Inverse Matrix inv(I-A)'
   dummy     'dummy objective'
;
```

```
equations
   inversedef(i,j) 'defines the inverse of (I-A)'
   obj             'dummy objective'
;

parameter ident(i,j);
ident(i,j)$sameas(i,j) = 1;

alias(k,j);
alias(kk,i);

* we want:
*    sum(k, inva(i,k)*(ident(k,j)-a(k,j))) =e= ident(i,j);
* but this gives a domain error, so we use a trick:
inversedef(i,j)..   sum(sameas(k,kk),
                        inva(i,k)*(ident(kk,j)-a(kk,j))) =e= ident(i,j);
obj..               dummy =e= 0;

model inv /inversedef,obj/;
solve inv using lp minimizing dummy;

display inva.l;


*---------------------------------------------------------------------
* calculate the Leontief Inverse matrix by a series.
* This is somewhat slow.
*---------------------------------------------------------------------

parameter Ak(i,j) 'A to the power k';
parameter Approx(i,j) 'current approximation of inv(I-A)';
parameter Temp(i,j);

set iter 'max number of iterations' /iter1*iter10/;
scalar converged /0/;
scalar maxelem;

*
* initalization
*
Ak(i,j) = A(i,j);
Approx(i,j) = Ident(i,j) + A(i,j);


loop(iter$(not converged),


* we want:
*    Temp(i,j) = sum(k, Ak(i,k)*A(k,j));
* but this gives a domain error, so we use a trick:

    Temp(i,j) = sum(sameas(k,kk), Ak(i,k)*A(kk,j));
    Ak(i,j) = Temp(i,j);

*
* converged?
*
    maxelem = smax((i,j),Ak(i,j));
    display maxelem;
    if (maxelem<0.00001, converged=1);

    Approx(i,j) = Approx(i,j) + Ak(i,j);


);


display Approx;

maxres = smax((i,j), abs(Approx(i,j)-InvA.l(i,j)));
display "max difference:",maxres;
```

6. An external equation solver

The GDX facility[4] allows for reading and writing data at run-time. We can use this to export a (non-singular) matrix $A = a_{i,j}$, let a program invert it, and load the inverse matrix A^{-1} back into GAMS.

invert1.gms. [9]

```
$ontext

   Finds the inverse of a matrix through an external program

   Erwin Kalvelagen, march 2005

   Reference: model gauss.gms from the model library
              http://www.gams.com/modlib/libhtml/gauss.htm

$offtext

set i  /i1*i3 /;
alias (i,j);

table a(i,j) 'original matrix'
      i1     i2     i3
i1    1      2      3
i2    1      3      4
i3    1      4      3
;

parameter inva(i,j) 'inverse of a';

execute_unload 'a.gdx',i,a;
execute '=invert.exe a.gdx i a b.gdx inva';
execute_load 'b.gdx',inva;

display a,inva;
```

The results look like:

```
----     28 PARAMETER a  original matrix

            i1          i2          i3

i1       1.000       2.000       3.000
i2       1.000       3.000       4.000
i3       1.000       4.000       3.000


----     28 PARAMETER inva  inverse of a

            i1          i2          i3

i1       3.500      -3.000       0.500
i2      -0.500                   0.500
i3      -0.500       1.000      -0.500
```

The algorithm used is an LU decomposition based linear solver routine from LAPACK [1] (routine DGESV). The wrapper is written in Fortran 90 and uses a GDX language binding described in [2].

The usage is described by running the program without arguments:

```
C:\gams projects\linear equations>invert
 INVERT: matrix inversion
 Usage
    > invert gdxin i a gdxout inva
```

[9] http://amsterdamoptimization.com/models/lineq/invert1.gms

```
 where
    gdxin  : name of gdxfile with matrix
    i      : name of set used in matrix
    a      : name of 2 dimensional parameter inside gdxin
    gdxout : name of gdxfile for results (eigenvalues)
    inva   : name of 2 dimensional parameter inside gdxout

 Calculates the inverse of a non-singular matrix a(i,j) where
 i and j are aliased sets. inva will contain the inverse
 matrix.


C:\gams projects\linear equations>
```

7. Eigenvalues and eigenvectors

A similar technique as used in the previous section can be used to calculate eigenvalues and eigenvectors of a symmetric matrix. A vector v is an eigenvector of a matrix A if

$$Av = \lambda v \tag{28}$$

or

$$(A - \lambda I)v = 0 \tag{29}$$

where λ is the corresponding eigenvalue. For non-symmetric matrices eigenvalues and eigenvectors are complex, but for a symmetric matrix they are real.

The external program `eigenvalue.exe` calculates just the eigenvalues, using routine DSYEV from LAPACK[1].

eigval.gms. [10]

```
$ontext

   Eigenvalue example.


octave:1> a=[9 1 1; 1 9 1; 1 1 9]
a =

  9  1  1
  1  9  1
  1  1  9

octave:2> eig(a)
ans =

   8
   8
  11


$offtext

set i /i1*i3/;
alias (i,j);

table a(i,j)
      i1   i2   i3
i1     9    1    1
i2     1    9    1
i3     1    1    9

;
```

[10] http://amsterdamoptimization.com/models/lineq/eigval.gms

```
parameter e(i) 'eigenvalues';

execute_unload 'mat.gdx',i,a;
execute '=eigenvalue.exe mat.gdx i a ev.gdx e';
execute_load 'ev.gdx',e;

display a,e;
```

The results are:

```
----     41 PARAMETER a

            i1          i2          i3

i1       9.000       1.000       1.000
i2       1.000       9.000       1.000
i3       1.000       1.000       9.000


----     41 PARAMETER e  eigenvalues

i1  8.000,    i2  8.000,    i3 11.000
```

To calculate both the eigenvalues and corresponding eigenvectors, the external program `eigenvector.exe` can be used. This program uses the same routine DSYEV from LAPACK[1].

eigvec.gms. [11]

```
$ontext

   Eigenvector example.


octave:1> a = [1 2 4 7 11; 2 3 5 8 12; 4 5 6 9 13; 7 8 9 10 14; 11 12 13 14 15]
a =

   1   2   4   7  11
   2   3   5   8  12
   4   5   6   9  13
   7   8   9  10  14
  11  12  13  14  15

octave:2> eig(a)
ans =

   -8.464425
   -1.116317
   -0.512109
   -0.027481
   45.120332

octave:3> [e1,e2] = eig(a)
e1 =

   0.5550905  -0.2642556   0.2892854   0.6748602   0.2879604
   0.4820641  -0.2581518   0.2196341  -0.7349311   0.3355726
   0.2865066   0.2159261  -0.8437897   0.0411896   0.3970041
  -0.0992784   0.7711236   0.3943678   0.0055409   0.4898525
  -0.6062562  -0.4714561  -0.0238286   0.0520829   0.6378888

e2 =

   -8.46442    0.00000    0.00000    0.00000    0.00000
    0.00000   -1.11632    0.00000    0.00000    0.00000
    0.00000    0.00000   -0.51211    0.00000    0.00000
    0.00000    0.00000    0.00000   -0.02748    0.00000
```

[11] http://amsterdamoptimization.com/models/lineq/eigvec.gms

```
    0.00000    0.00000    0.00000    0.00000   45.12033

$offtext

set i /i1*i5/;
alias (i,j);

table a(i,j)
      i1   i2   i3   i4   i5
i1     1    2    4    7   11
i2     2    3    5    8   12
i3     4    5    6    9   13
i4     7    8    9   10   14
i5    11   12   13   14   15

;

parameter eval(i) 'eigenvalues';
parameter evec(i,j) 'eigenvectors';

execute_unload 'mat.gdx',i,a;
execute '=eigenvector.exe mat.gdx i a ev.gdx eval evec';
execute_load 'ev.gdx',eval, evec;

display a, eval, evec;
```

The results are:

```
----     64 PARAMETER a

            i1          i2          i3          i4          i5

i1       1.000       2.000       4.000       7.000      11.000
i2       2.000       3.000       5.000       8.000      12.000
i3       4.000       5.000       6.000       9.000      13.000
i4       7.000       8.000       9.000      10.000      14.000
i5      11.000      12.000      13.000      14.000      15.000


----     64 PARAMETER eval  eigenvalues

i1 -8.464,    i2 -1.116,    i3 -0.512,    i4 -0.027,    i5 45.120


----     64 PARAMETER evec  eigenvectors

            i1          i2          i3          i4          i5

i1       0.555      -0.264       0.289       0.675       0.288
i2       0.482      -0.258       0.220      -0.735       0.336
i3       0.287       0.216      -0.844       0.041       0.397
i4      -0.099       0.771       0.394       0.006       0.490
i5      -0.606      -0.471      -0.024       0.052       0.638
```

References

1. E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK users' guide*, third ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.
2. Erwin Kalvelagen, *Interfacing GAMS with other applications; tutorial and examples*.
3. Statistics Bureau Japan, *1995 Input-Output Tables for Japan*, Tech. report, Statistical Standards Department, Statistics Bureau, Management and Coordination Agency, March 2000.
4. Paul van der Eijk, *GDX facilities in GAMS*.

Amsterdam Optimization Modeling Group LLC
E-mail address: erwin@amsterdamoptimization.com