Empirical Analysis of the Role of Energy in Economic Growth

Caleb Reese[a], Lucas Timmer[a], Matthew Kuperus Heun[a,*]

[a] *Engineering Department, Calvin College, Grand Rapids, MI 49546, USA*

Abstract

*********** Add abstract ***********

Keywords: economic growth, energy, cobb-douglas, CES, LINEX

Caleb, put your LaTeX code here.

1. Cobb-Douglas Without Energy

Table 1: Cobb-Douglas parameters for 1980-2011 (US, UK, JP) or 1991-2011 (others). (Parameter estimates beneath symbol. 95% confidence bounds to left and right.)

		λ			α			β	
US	0.0087	0.0102	0.0116	0.21	0.27	0.34	0.66	0.73	0.79
UK	-0.0104	0.0097	0.0303	-0.25	0.44	1.12	-0.13	0.56	1.24
JP	0.0021	0.0052	0.0082	0.44	0.52	0.59	0.41	0.48	0.56
CN	-0.0405	0.0188	0.0779	0.11	0.71	1.32	-0.32	0.29	0.89
ZA	-0.0007	0.0008	0.0022	0.46	0.60	0.73	0.26	0.40	0.54
SA	-0.0159	-0.0123	-0.0087	0.21	0.45	0.68	0.32	0.55	0.78
IR	0.0032	0.0039	0.0045	0.49	0.60	0.70	0.30	0.40	0.51
TZ	-0.0039	0.0015	0.0068	0.50	0.73	0.95	0.05	0.27	0.50
ZM	0.0218	0.0249	0.0280	1.25	1.41	1.57	-0.57	-0.41	-0.25

*Corresponding author

Email address: mkh2@calvin.edu, tel: +1 (616) 526-6663, fax: +1 (616) 526-6501 (Matthew Kuperus Heun)

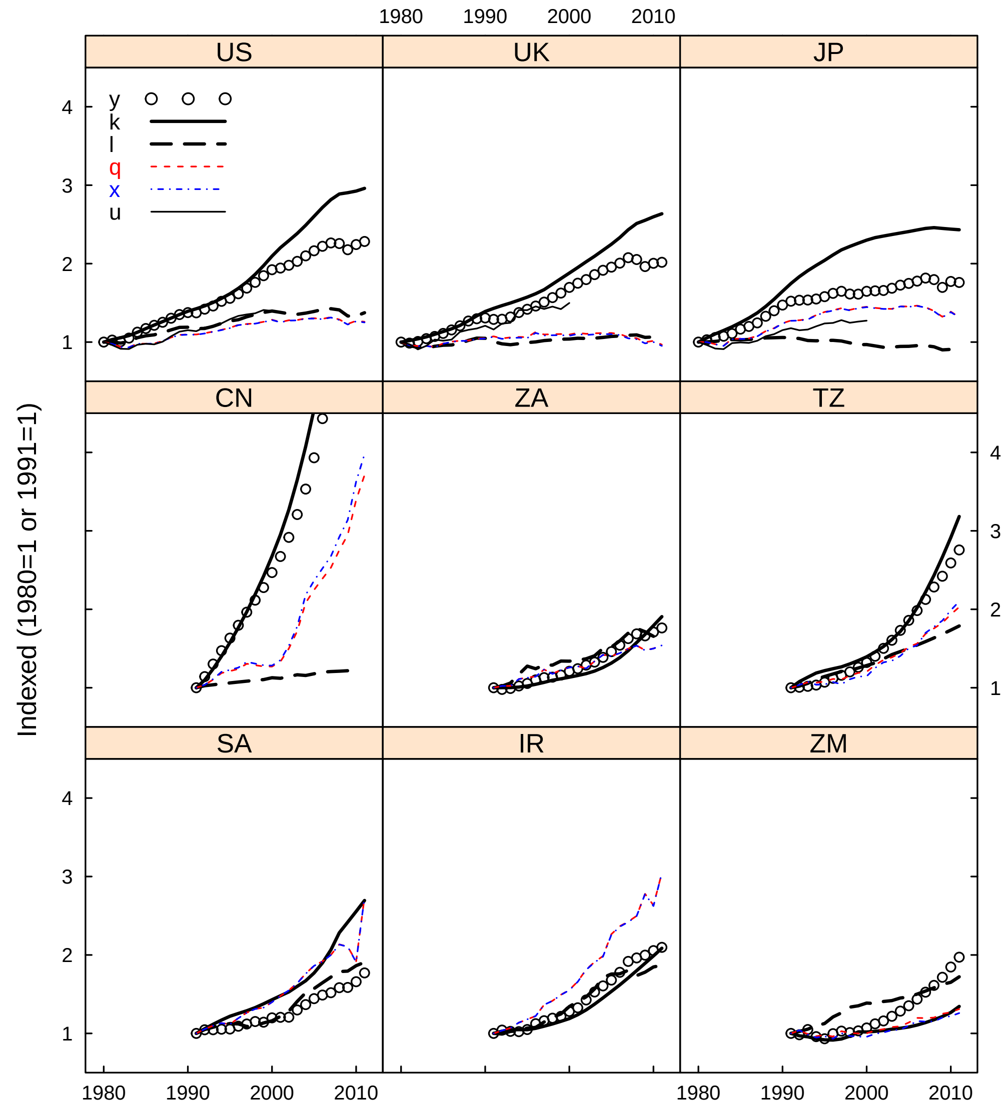

Figure 1: GDP (y), capital stock (k), labor (l), thermal energy (q), exergy (x), and useful work (u) for all economies. (China's indexed GDP and indexed capital stock rise to $y = 7.3$ and $k = 9.2$ in 2011.)

```r
usModel <- cobbDouglasModel("US")
# print(coef(usModel))
usPred <- predict(usModel) #See http://stackoverflow.com/questions/9918807/how-get
#print(usPred)
data.frame(usPred)
```

```
   usPred
1   1.000
2   1.021
3   1.026
4   1.057
5   1.119
6   1.162
7   1.196
8   1.244
9   1.296
10  1.348
11  1.375
12  1.383
13  1.408
14  1.458
15  1.520
16  1.579
17  1.628
18  1.702
19  1.772
20  1.844
21  1.912
22  1.941
23  1.963
24  1.997
25  2.058
26  2.128
27  2.204
28  2.260
29  2.281
30  2.215
31  2.241
```

```
32  2.323
```

```
# allData <- loadData("All")
# predictions <- cobbDouglasPredictionsColumn()
# allData <- cbind(allData, predictions)
#
# print(allData)
```

```
    predGDP
1    1.0000
2    1.0206
3    1.0260
4    1.0574
5    1.1188
6    1.1617
7    1.1960
8    1.2438
9    1.2959
10   1.3479
11   1.3747
12   1.3831
13   1.4078
14   1.4579
15   1.5203
16   1.5790
17   1.6283
18   1.7016
19   1.7716
20   1.8440
21   1.9118
22   1.9406
23   1.9632
24   1.9971
25   2.0579
26   2.1281
27   2.2039
```

```
28   2.2598
29   2.2806
30   2.2146
31   2.2410
32   2.3234
33   1.0000
34   0.9917
35   0.9988
36   1.0134
37   1.0516
38   1.0850
39   1.1125
40   1.1559
41   1.2165
42   1.2748
43   1.3089
44   1.3079
45   1.3146
46   1.3321
47   1.3702
48   1.4093
49   1.4477
50   1.4962
51   1.5457
52   1.5965
53   1.6418
54   1.6943
55   1.7365
56   1.7840
57   1.8393
58   1.8986
59   1.9551
60   2.0204
61   2.0744
62   2.0753
63   2.1175
64   2.1507
```

65	1.0000
66	1.0341
67	1.0675
68	1.1023
69	1.1362
70	1.1669
71	1.2030
72	1.2440
73	1.2962
74	1.3463
75	1.4004
76	1.4489
77	1.4862
78	1.5076
79	1.5401
80	1.5772
81	1.6141
82	1.6410
83	1.6477
84	1.6528
85	1.6774
86	1.6836
87	1.6845
88	1.7023
89	1.7261
90	1.7432
91	1.7679
92	1.7838
93	1.7840
94	1.7526
95	1.7641
96	1.7440
97	1.0000
98	1.0975
99	1.2235
100	1.3647
101	1.5167

```
102  1.6811
103  1.8553
104  2.0466
105  2.2459
106  2.4723
107  2.6961
108  2.9711
109  3.2927
110  3.6207
111  4.0117
112  4.4483
113  4.9207
114  5.4276
115  6.0583
116  6.7711
117  1.0000
118  1.0099
119  1.0264
120  1.0732
121  1.1213
122  1.1238
123  1.1582
124  1.1761
125  1.2075
126  1.2210
127  1.2402
128  1.2634
129  1.2996
130  1.3643
131  1.4082
132  1.4839
133  1.5783
134  1.6682
135  1.7175
136  1.7616
137  1.0000
138  1.0411
```

```
139  1.0691
140  1.0932
141  1.1105
142  1.1084
143  1.0905
144  1.0967
145  1.1211
146  1.1406
147  1.1748
148  1.2125
149  1.2863
150  1.3501
151  1.3953
152  1.4590
153  1.5323
154  1.6186
155  1.6435
156  1.6990
157  1.7406
158  1.0000
159  1.0178
160  1.0454
161  1.0538
162  1.0717
163  1.0899
164  1.1390
165  1.1857
166  1.2278
167  1.2926
168  1.3538
169  1.4234
170  1.5165
171  1.6318
172  1.7109
173  1.7735
174  1.8569
175  1.8944
```

```
176  1.9799
177  2.0761
178  2.1548
179  1.0000
180  1.0639
181  1.1155
182  1.1713
183  1.2042
184  1.2333
185  1.2610
186  1.2943
187  1.3360
188  1.3816
189  1.4377
190  1.5055
191  1.5879
192  1.6755
193  1.7869
194  1.9203
195  2.0749
196  2.2377
197  2.4150
198  2.6023
199  2.8013
200  1.0000
201  0.9708
202  0.9669
203  0.9380
204  0.9327
205  0.9256
206  0.9514
207  1.0057
208  1.1091
209  1.1221
210  1.1656
211  1.2052
212  1.2585
```

```
213  1.2987
214  1.3548
215  1.4198
216  1.4976
217  1.5886
218  1.6883
219  1.8248
220  1.9983

Error:  arguments imply differing number of rows:  222, 220
```

2. Cobb-Douglas With Energy

We can force α, β, and γ to be in $[0, 1]$ by a reparameterization:

$$a \in [0, 1], b \in [0, 1], \alpha = \min(a, b), \beta = |b - a|, \gamma = 1 - \max(a, b)$$

2.1. Cobb-Douglas with Q

```
# Note that the anlaysis of ZA is taking a long time here. Need to figure out why.
CDqTables <- lapply(countries, cobbDouglasEnergyTable, energyType="Q")
```

```
print(CDqTables[["US"]], caption.placement="top")
print(CDqTables[["ZA"]], caption.placement="top")
# According to http://cran.r-project.org/web/packages/xtable/vignettes/xtableGalle
# be able to use the "sanitize.text.function" parameter to allow markup in column
# line is not working at the present time. --MKH, 18 Jan 2012.
# print(tableCDe, sanitize.text.function = function(x){x})

#print(tableAll, caption.placement="top")
```

2.2. Cobb-Douglas With X

```
# Note that the anlaysis of ZA is taking a long time here. Need to figure out why.
CDxTables <- lapply(countries, cobbDouglasEnergyTable, energyType="X")
```

```
print(CDxTables[["US"]], caption.placement="top")
print(CDxTables[["ZA"]], caption.placement="top")
```

2.3. Cobb-Douglas With U

```
CDuTables <- lapply(countries, cobbDouglasEnergyTable, energyType="U")
```

```
print(CDuTables[["US"]], caption.placement="top")
print(CDuTables[["ZA"]], caption.placement="top")
```

3. CES

```
cesData <- function(countryName, energyType){
  energyColumnName <- paste("i", energyType, sep="")
  # Load the data that we need.
  dataTable <- loadData(countryName)

  # Establish guess values for phi beta, zeta, lambda_L and lambda_E.
  phiGuess <- -20
  betaGuess <- 0.5 # a typical value for beta (exponent on labor)
  zetaGuess <- 0.0004 # a small value
  lambda_LGuess <- 0.007 #assuming no technical progress on the labor-capital port
  lambda_EGuess <- 0.008 #assuming no technical progress on the energy portion of

  # Runs a non-linear least squares fit to the data with constraints
  modelCES <- nls(iGDP ~ ((1-zeta) * (exp(lambda_L*iYear) * iCapStk^(1-beta) * iLa
                         + zeta*(exp(lambda_E*iYear) * iQ)^phi)^(1/phi),
                  algorithm = "port",
                  control = nls.control(maxiter = 500, tol = 1e-06, minFactor = 1
                                        printEval = FALSE, warnOnly = FALSE),
```

```
                 start = list(phi=phiGuess, beta=betaGuess, zeta=zetaGuess, lamb
                              lambda_E=lambda_EGuess),
                 lower = list(phi=-Inf, beta=0, zeta=0, lambda_L=-Inf, lambda_E=
                 upper = list(phi=0, beta=1, zeta=1, lambda_L=Inf, lambda_E=Inf)
                 data=dataTable)

aicCES <- AIC(modelCES, k=2) # Checks validity of the model. AIC stands for Akai
print(aicCES)

# Gives the nls summary table
summaryCES <- summary(modelCES) # Gives the nls summary table
print(summaryCES)

# Provides confidence intervals on phi, beta, zeta, lambda_L, and lambda_E. But,
ciCES <- confint(modelCES, level = ciLevel)
print(ciCES)

# Get the estimate for alpha
beta <- as.numeric(coef(modelCES)["beta"])
alpha <- 1.0 - beta
alpha.est <- deltaMethod(modelCES, "1 - beta") # Estimates alpha and its standar
print(alpha.est)

# Now calculate a confidence interval on alpha
dofCES <- summaryCES$df[2]
print(dofCES) # Gives the degrees of freedom for the model.
tvalCES <- qt(ciHalfLevel, df = dofCES); tvalCES
# Get confidence intervals for each parameter in the model
alphaCICES <- with(alpha.est, Estimate + c(-1.0, 1.0) * tvalCES * SE) # CI on al
print(alphaCICES)

# Assemble the data into data frames for the table.
estCES <- data.frame(phi = coef(modelCES)["phi"], alpha = alpha,
                     beta = coef(modelCES)["beta"], zeta = coef(modelCES)["zeta'
                     lambda_L = coef(modelCES)["lambda_L"], lambda_E = coef(mode
row.names(estCES) <- paste("CES with ", energyType, sep="")
#print(estCES)
```

```
  # The [1] subscripts pick off the lower confidence interval
  lowerCES <- data.frame(phi = ciCES["phi","2.5%"], alpha = alphaCICES[1],
                         beta = ciCES["beta", "2.5%"], zeta = ciCES["zeta", "2.5%'
                         lambda_L = ciCES["lambda_L", "2.5%"], lambda_E = ciCES["l
  row.names(lowerCES) <- "- 95% CI"
  # The [2] subscripts pick off the lower confidence interval
  upperCES <- data.frame(phi = ciCES["phi","97.5%"], alpha = alphaCICES[2],
                         beta = ciCES["beta", "97.5%"], zeta = ciCES["zeta", "97.5
                         lambda_L = ciCES["lambda_L", "97.5%"], lambda_E = ciCES["
  row.names(upperCES) <- "+ 95% CI"

  # Now create the data for a table.
  dataCES <- rbind(upperCES, estCES, lowerCES)
  print(dataCES)
  return(dataCES)

  #xyplot( resid(modelCESQ) ~ fitted(modelCESQ) )
  #histogram( ~resid(modelCESQ) )
  #qqmath( ~resid(modelCESQ) )
}

####################################
# Creates a LaTeX printable table from the CES data. This function first calls ces
#
# countryName is a string containint the 2-letter abbreviation for the country, e.
# energyType is a string to be used in table captions reprsenting the type of ener
#
# returns a printable LaTeX table from xtable.
##
cesTable <- function(countryName, energyType){
  dataCESe <- cesData(countryName, energyType)
  tableCESq <- xtable(dataCESe, caption=paste(countryName, ", 1980-2011.", sep=""))
}
```

3.1. CES with Q

```
countryName <- "US"
energyType <- "Q"
tableCESq <- cesTable(countryName, energyType)
```

```
[1] -194

Formula: iGDP ~ ((1 - zeta) * (exp(lambda_L * iYear) * iCapStk^(1 - beta) *
    iLabor^beta)^phi + zeta * (exp(lambda_E * iYear) * iQ)^phi)^(1/phi)

Parameters:
          Estimate Std. Error t value Pr(>|t|)
phi      -3.96e+01   2.43e+01   -1.63   0.1144
beta      6.09e-01   3.45e-02   17.64  2.4e-16
zeta      2.09e-06   1.32e-05    0.16   0.8758
lambda_L  7.98e-03   6.68e-04   11.95  2.8e-12
lambda_E  8.57e-03   2.48e-03    3.45   0.0018

Residual standard error: 0.0105 on 27 degrees of freedom

Algorithm "port", convergence message: relative convergence (4)
```

```
Waiting for profiling to be done...
```

```
             2.5%      97.5%
phi            NA -10.290831
beta     0.514667   0.665371
zeta           NA         NA
lambda_L 0.006428   0.009152
lambda_E 0.000715   0.012468
         Estimate      SE
1 - beta   0.3911 0.03453
[1] 27
[1] 0.3202 0.4619
              phi  alpha   beta      zeta lambda_L lambda_E
+ 95% CI   -10.29 0.4619 0.6654        NA 0.009152 0.012468
CES with Q -39.64 0.3911 0.6089 2.085e-06 0.007979 0.008570
- 95% CI       NA 0.3202 0.5147        NA 0.006428 0.000715
```

```
#CESqTables <- lapply(countries, cesTable, energyType="Q")
```

```r
print(tableCESq, caption.placement="top")
```

Table 2: US, 1980-2011.

	phi	alpha	beta	zeta	lambda_L	lambda_E
+ 95% CI	-10.3	0.46	0.67		0.00915	0.01247
CES with Q	-39.6	0.39	0.61	0.000002	0.00798	0.00857
- 95% CI		0.32	0.51		0.00643	0.00071

```r
#print(CESqTables[["US"]], caption.placement="top")
#print(CESqTables[["ZA"]], caption.placement="top")
```