# Instructions / Code for Ossa – Trice Model Calculations

*Paul Sztorc*[1]

*October 7, 2014*

This document contains the instructions and the code for doing the runs with the Ossa model that are used to calibrate the TRICE model. It starts with a description and then contains the code for the runs.

## Instructions:

Using: MATLAB R2014a (8.3.0.532) 64-bit
…in a working directory containing Ossa's files.
…run "Set 1" (model/Main_RunMe.m)
Then, in the same directory,
Using: RStudio 0.98.1049 using R 3.0.1 (64-bit)
…run "Set 2" (model/results/r_main.Rmd) .

Set 1 includes calculations run by Ossa himself ("mycalculations"), which provide the model-setup. From here, I set the tariff shocks with "Shocks = (0:10)/100", which produces shocks of 0, .01, .02, … , .10 . Next, I create blank MATLAB arrays in which to store the results of the program. Finally, I compute the number of possible clubs, and loop through each club. For each club, I have all of those countries who are 'In the Club' each increase their tariffs on those countries who are 'Out of the Club', and record the impact this has on each country's Welfare. Finally, I save the results to .mat files.

Set 2 contains R code written entirely to interpret and graph the outputs of Set 1. The first block of code defines some basic user-friendliness functions and loads some external code from the widely-available CRAN network. Secondly, I load exogenous information: which country-labels were used, how countries are to be weighted, and which tariff-shocks we selected. The third block attempts to reconstruct the original baseline tariffs for each country. The fourth block loads the .mat files from Set 1, and reshapes them into 'long form', and deposits them into .csv files for easy use by other software packages. The fifth block selects each country's optimal tariff (ie, the row which contains the highest 'Welfare' value).

[1] Paul Sztorc is Associate in Research in the Economics Department, Yale University. This program provides the background for the calibrations used in William Nordhaus's TRICE model, forthcoming, American Economic Review, 2015.

The sixth block of code in Set 2 produces .pdf graphs of a countries welfare-response (y axis) to each tariff shock (x-axis). Each page is a different club size: on page 1, clubs are of one country, meaning that there are 7 ways of being in the club and 6*7 = 42 ways of being out of the club (6 "out countries" per each of the 7 clubs). The seventh block calculates the optimal tariff of each country, for a given club size, by fitting a no-intercept polynomial regression ( (y-1) = 0 + B*tariff + (B^2) * tariff ), and then performing simple calculus. The resulting .csv files ("InTheClub.csv" and "OutOfClub.csv") contain the optimal shock, the welfare which that shock achieves ('WelfareStar'), as well as, separately, the slope of each regression at a 5% shock level.

The eighth block of code in Set 2 uses the exogenous weights defined in block 2 (above) to calculate the welfare of various aggregate regions: The welfare of a country which is in a club by itself, the weighted average welfare of the 6 countries which are out, and the global weighted average welfare of all 7 countries. The ninth and final code block simply writes a .csv file of Ossa's tariff data, without aggregating or editing it in any way, so that one might aggregate the industry tariffs (of which there are 33) in a different way (or not at all).

## Set 1 (MATLAB):

```
%----------------------
% OSSA Model
%----------------------

clear all
close all
clc

mycalculations %This program performs a number of frequently needed calculations
TRADEs %TRADEs(i,j,s) is the value of trade flowing from country i to country j in industry s; see regions.csv and sectors.csv for the list of regions and industries
TARIFFs %TARIFFs(i,j,s) is the tariff applied by country j against industry s imports from country i
SIGMA %These are the elasticities of substitution from Table 1 of the paper
NX %These are aggregate net exports which are basically zero indicating that this is the purged version of the raw data (as explained in the paper)

%--------------------------------------------------
%Computing the effects of exogenous tariff changes
%--------------------------------------------------

clear all
close all
clc


% Shocks = (0:10)/100;
 Shocks = (25:65)/100;
% Shocks = [0.5014162, 0.5330983, 0.5408727, 0.4320262, 0.5654997, 0.5451430, 0.5540263];
```

```
L = length(Shocks);
C = 7; %seven countries/regions


% 1 Brazil
% 2 China
% 3 EU
% 4 India
% 5 Japan
% 6 ROW
% 7 US



mycalculations

% create a place to dump the results
BigResults = zeros(C,L,35,(C-1));  %35 happens to be max { rows( nchoosek(1:7,1:7) ) }
BigWhoIsIn = ones(35,C,(C-1)) * -1; % later, we will remove all -1's;

% separately, keep all the tarrifs
dimTARIFFs = size(TARIFFs);
AllTariffs = zeros(dimTARIFFs(1),dimTARIFFs(2),dimTARIFFs(3),L);

%for ClubSize = 1 % special edit - for simplicity (when only considering
%club sizes of one

for ClubSize = 1:(C-1) %clubs can be all or one

   ClubsOfThisClubSize = nchoosek(1:C,ClubSize);
   dimClubs = size(ClubsOfThisClubSize); %rows and columns of these clubs
   qClubs = dimClubs(1);

   WhoIsIn = zeros(qClubs,C); % declare for use - all out

   for ClubIndex = 1:qClubs
      Club = ClubsOfThisClubSize(ClubIndex,:);

      % Keep track of who is in
      for n = 1:C % ...for each "i" trading partner...
         if ismember(n, Club)
            WhoIsIn(ClubIndex,n) = 1;
         end
      end

      BigWhoIsIn(ClubIndex,:,ClubSize) = WhoIsIn(ClubIndex,:);

      for Shock = 1:L   %for each shock...
```

```
        %have the importer shock the other countries by increasing their
        %tarrifs by Shocks(n)
        TARIFFCs=TARIFFs;  %reset tarrifs

        for n = 1:C % ...for each "i" trading partner...
           if not(ismember(n, Club)) %If they aren't in the climate club, they're getting taxed (rows)
              for o = 1:C %...for each importer...
                 if ismember(o, Club) %only club-members are doing the taxing (columns)
                    % TARIFFCs(n,o,:)=TARIFFCs(n,o,:)+Shocks(Shock);
                    % % additive method
                    TARIFFCs(n,o,:)= (1+TARIFFCs(n,o,:))*(1+Shocks(Shock))-1;

                 end
              end
           end
        end


     %welfare calculations
     NXC=zeros(N,1); %NXC are counterfactual aggregate net exports. I use this to purge the raw data from aggregate trade imbalances as described in the main text
     LAMBDA=LAMBDABAS; %Select LAMBDABAS if you don't want the lobbying weights, and LAMBDAPOL otherwise

[GOVERNMENTWELFAREHAT,WELFAREHAT,WAGEHAT,TRADECs,LOBBYWELFAREHAT,EXPENDITUREHAT]=mycounterfactuals(TARIFFCs,NXC,LAMBDA);
     BigResults(:,Shock,ClubIndex,ClubSize) = WELFAREHAT;  %country-welfare-rows, tarrif-effects (columns), by The Club Itself, ClubSize

     % save also the tarrifs
     AllTariffs(:,:,:,Shock) = TARIFFCs;

     end
  end



end

save('results\bigresults.mat', 'BigResults') ;

save('results\shocks.mat', 'Shocks') ;

save('results\basetariffs.mat', 'TARIFFs') ;

save('results\alltariffs.mat', 'AllTariffs') ;

save('results\whoisin.mat', 'BigWhoIsIn') ;
```

## Set 2 (R):

Ossa Model ( Data for TRICE )
==========

Paul Sztorc
`r date()`
.Rmd R markdown file.
Written in R using version 3.0.1
Made with RStudio 0.98.1049

```{r PreLoad,echo=FALSE,message=FALSE}

rm(list=ls())

Use <- function(package) {
  if(suppressWarnings(!require(package,character.only=TRUE))) install.packages(package,repos="http://cran.case.edu/")
  require(package,character.only=TRUE)
}

Pst <- function(...) paste(...,sep="")


setwd("C:/Users/ps583/Documents/GitHub/TRICE/model/results")

Use('R.matlab')
Use('reshape')
Use('ggplot2')

```

This is an R Markdown document.

```{r ExogenousLabelsWeightsShocks}

#Label Countries
DFlabs <- read.csv("regions.csv")
names(DFlabs) <- c("country","importer")

Weights <- c(.0279, .1540, .121, .0583, .0540, .3917, .1931) # estimate of gdp weights - exogenous
names(Weights) <- DFlabs$importer

# Which shocks did we use in Matlab?
Shocks <- as.vector(readMat("shocks.mat")$Shocks)

Suffix <- Pst("_", round(Shocks[which.min(Shocks)]*100,2), "_", round(Shocks[which.max(Shocks)]*100,2), ".csv")

```

```{r OriginalTarrifs}
```

```
# A Completely Optional Step for Looking at the unperturbed Tarif

# First,
# Import and format tarriff data
OriginalTarrifs <- readMat("basetariffs.mat")
mOriginalTarrifs <- melt(OriginalTarrifs)[,-5] # lose a useless column
names(mOriginalTarrifs) <- c("ExporterGettingTaxed","ImporterApplyingTax","Industry","Tarrif")

SidewaysTarrifs <- cast(data=mOriginalTarrifs,formula=ExporterGettingTaxed~ImporterApplyingTax+Industry)
AverageTarrifs <- cast(data=mOriginalTarrifs,formula=ExporterGettingTaxed~ImporterApplyingTax,fun.aggregate=mean)

# # Optional:
# write.csv(mOriginalTarrifs,file="original_tarrifs.csv")
# write.csv(SidewaysTarrifs,file="original_tarrifs_stacked_sideways.csv")
# write.csv(AverageTarrifs,file="original_tarrifs_avg_by_industry.csv")

# average tarrif
ResultsOT <- data.frame("country"=1:7, "importer"=DFlabs$importer, "origTarrif"=NA)
for(i in 1:7) {
  TempAT <- as.matrix(AverageTarrifs)[-i,i]
  TempWeights <- Weights[-i]
  TempWeights <- TempWeights / sum(TempWeights)
  TempRes <- TempAT %*% TempWeights
  ResultsOT$origTarrif[i] <- TempRes
}

# write.csv(ResultsOT,file="original_tarrifs_avg_by_industry_wgt_by_GDP.csv")

```


```{r LoadMatlabResults}

MatData <- readMat("bigresults.mat")
MatClubMembership <- readMat("whoisin.mat")

# Shape into useful form
mDF <- melt(MatData)
mDF <- mDF[mDF$value!=0,-6]  # remove stuff which never should have been there
names(mDF) <- c("country","shock","club","clubsize","welfare")

mDF <- merge(mDF,DFlabs)
mDF$cgroup <- paste(mDF$country,mDF$club,sep=".") # which version of the club are we in
mDF$clubindex <- paste(mDF$clubsize,mDF$club,sep=":")

# Add in the actual shocks - more clear
mDF  <- merge( mDF, data.frame( "rawshock"=Shocks,"shock"=(1:length(Shocks)) ) )
```

```
# Club Membership

mCM <- melt(MatClubMembership)
mCM <- mCM[mCM$value!=-1,-5]  # remove stuff which never should have been there

# normal names
names(mCM) <- c("club", "country", "clubsize","InTheClub")
ClubMembers <- cast(mCM, formula = clubsize + club ~ country, value = "InTheClub")

# label the countries
names(ClubMembers) <- c("clubsize","club",levels(DFlabs$importer))
ClubMembers$clubindex <- paste(ClubMembers$clubsize,ClubMembers$club,sep=":")

# Sanity Check
ClubMembers

Huge <- merge(ClubMembers,mDF)

head(Huge)


write.csv(mDF,Pst("MatlabOutput",Suffix))
write.csv(Huge,Pst("AnnotatedMatlabOutput",Suffix),row.names=FALSE)

```

```{r MaxCalculatedNumerically}
# Maxes

# Maxes <- mDF[mDF$ClubStatus=="In"&mDF$clubsize==2,] # Club only

Maxes <- mDF[mDF$clubsize==1,] # Club only

MaxesCast <- cast(Maxes,fun.aggregate = max, formula = importer ~ ., value = "welfare"  ) # get max row only.

# remerge with old data
names(MaxesCast) <- c("importer","welfare")
MaxesFull <- merge( MaxesCast, mDF[, c("welfare", "cgroup", "rawshock")] )

write.csv(MaxesFull, Pst("MaxTarrifsFromOssa",Suffix))

```

```{r Plots}

# Graphical Representation of Trade Data

# Fix Suffix (for files later) - must end in '.pdf', of course
```

```
SuffixPDF <- paste( strsplit(Suffix,".",fixed = TRUE)[[1]][1], ".pdf", sep="")

Plots <- vector("list",6)
for( i in 1:6 ) {  # for each club size

  # Subset the Data
  Slice <- mDF[mDF$clubsize==i,]

  # Build the Plot
  Plots[[i]] <- ggplot(Slice ,aes(y=welfare,x=rawshock,colour=importer)) +
    geom_point(size=.5) +
    geom_line(aes(group=cgroup),alpha=.2) +
    labs(title=paste("Clubs of size",i))
}

# Write to File
pdf(file=Pst("AllNations",SuffixPDF))
for( i in 1:6 ) print(Plots[[i]])
dev.off()


Plots <- vector("list",6)
for( i in 1:6 ) {  # for each club size

  # Subset the Data
  Slice <- mDF[mDF$clubsize==i,]
  ClubOnly <- Slice[Slice$welfare >= 1, ] # this happens to be always correct (and graphically what we are interested in, anyway)

  # Make the Plot
  Plots[[i]] <- ggplot(ClubOnly ,aes(y=welfare,x=rawshock,colour=importer)) +
    geom_point(size=.5) +
    geom_line(aes(group=cgroup),alpha=.2) +
    stat_smooth(aes(fill=importer), method="lm",formula = y~poly(x,2,raw=TRUE)) +
    labs(title=paste("Clubs of size",i))
}

pdf(file=Pst("ClubNationsOnly",SuffixPDF))
for( i in 1:6 ) print(Plots[[i]])
dev.off()

Plots <- vector("list",6)
for( i in 1:6 ) {  # for each club size

  # Subset the Data
  Slice <- mDF[mDF$clubsize==i,]
  NonClubOnly <- Slice[Slice$welfare <= 1, ]

  # Make the Plot
  Plots[[i]] <- ggplot(NonClubOnly ,aes(y=welfare,x=rawshock,colour=importer)) +
```

```
    geom_point(size=.5) +
    geom_line(aes(group=cgroup),alpha=.2) +
    stat_smooth(aes(fill=importer), method="lm",formula = y~poly(x,2,raw=TRUE)) +
    labs(title=paste("Clubs of size",i))
}

pdf(file=Pst("NonClubOnly",SuffixPDF))
for( i in 1:6 ) print(Plots[[i]])
dev.off()

```


```{r OptimalTarrifs}
# Calculate Optimal Tarrif and Slope at 10%


# get ready to merge this info
ShockDf <- data.frame(shock=1:length(Shocks),rawshock=Shocks)
LargeDf <- merge(mDF,ShockDf)



for( i in unique( LargeDf$clubsize ) ) {  # for each club size

  # Get the data points
  Slice <- LargeDf[LargeDf$clubsize==i,]
  N <- nrow(Slice)

  # Partition by Club-membership
  ClubOnly <- Slice[Slice$welfare >= 1, ]
  NonClubOnly <- Slice[Slice$welfare <= 1, ]

  # Models - NO INTERCEPT
  m1 <- lm( I(welfare-1) ~ rawshock:importer+I(rawshock^2):importer + 0, data=ClubOnly)  # I (y -1) forces origin to be at 0,0
  m2 <- lm( I(welfare-1) ~ rawshock:importer+I(rawshock^2):importer + 0, data=NonClubOnly)

  ThisRowM1 <- data.frame("ClubSize"=rep(i,7),
                    "importer"=DFlabs$importer,
                    "xBeta"=matrix(coef(m1),ncol=2)[,1],
                    "x2Beta"=matrix(coef(m1),ncol=2)[,2],
                    "df"=summary(m1)$df[2],
                    "r2"= summary(m1)$r.squared )

  ThisRowM2 <- data.frame("ClubSize"=rep(i,7),
                     "importer"=DFlabs$importer,
                     "xBeta"=matrix(coef(m2),ncol=2)[,1],
                     "x2Beta"=matrix(coef(m2),ncol=2)[,2],
                     "df"=summary(m2)$df[2],
```

```
                    "r2"= summary(m2)$r.squared )

  # Create, then append the data:

  # are we first?
  FirstRow <- i==unique( LargeDf$clubsize )[1]
  if(FirstRow) InDF <- ThisRowM1
  if(!FirstRow) InDF <- rbind(InDF, ThisRowM1)

  if(FirstRow) OutDF <- ThisRowM2
  if(!FirstRow) OutDF <- rbind(OutDF, ThisRowM2)
}



# Simple slope calculation - first derivative
InDF$SlopeAtTen <-  ( InDF$xBeta +  (2*InDF$x2Beta*.1))     # where x=.1, what is the slope ?
OutDF$SlopeAtTen <- ( OutDF$xBeta + (2*OutDF$x2Beta*.1))

# Basic Calculus-based optimization
InDF$OptShock <-    -InDF$xBeta /  (2*InDF$x2Beta)
OutDF$PessShock <-  -OutDF$xBeta / (2*OutDF$x2Beta)

# Basic Calculus-based optimization
InDF$OptShock <-    -InDF$xBeta /  (2*InDF$x2Beta)
OutDF$PessShock <-  -OutDF$xBeta / (2*OutDF$x2Beta)

# Calculate the Actual Optimized Welfare
InDF$WelfareStar <-    InDF$xBeta*InDF$OptShock + InDF$x2Beta*InDF$OptShock*InDF$OptShock + 1  # we originally
subtracted 1
OutDF$WelfareStar <-  OutDF$xBeta*OutDF$PessShock + OutDF$x2Beta*OutDF$PessShock*OutDF$PessShock + 1

# Special Request: Value at 5 %
# Calculate the Actual Optimized Welfare
InDF$Welfare5pct <-    InDF$xBeta*0.05 + InDF$x2Beta*0.05*0.05 + 1  # we originally subtracted 1
OutDF$Welfare5pct <-  OutDF$xBeta*0.05 + OutDF$x2Beta*0.05*0.05 + 1

# Dump results
write.csv( InDF, file=Pst("InTheClub",Suffix) )
write.csv( OutDF,file=Pst("InTheClub",Suffix) )

```

```
```{r GlobalWelfare}


OssaTables <- vector("list",length = length(Shocks) )
```
```

```
for(Shock in Shocks ) { # for each importer


 # # Global Welfare # #

 TempDf <- LargeDf[LargeDf$clubsize==1 & LargeDf$rawshock==Shock,] # at 10% shock  # at 60%

 # "In" clubs of one
 OssaTable <- TempDf[TempDf$country==TempDf$club,c("welfare","importer"),][,c(2,1)]
 OssaTable <- merge(OssaTable,DFlabs)
 names(OssaTable) <- c("country","InWelfare",'country')

 # slightly more complicated subset...produces many results needs aggregation
 Out <- TempDf[TempDf$country!=TempDf$club,]


 # Other countries - slightly complex
 Results <- vector(length=7)
 for(i in 1:7) {
  TempWel <- Out[Out$country==i,"welfare"]
  TempWeights <- Weights[-i]
  TempWeights <- TempWeights / sum(TempWeights)
  TempRes <- TempWel %*% TempWeights
  Results[i] <- TempRes
  }
 OssaTable$ElseWelfare <- Results


 # World - very easy, just weighted average (multiply)
 Results2 <- vector(length=7)
 for(i in 1:7) {
  TempWel <- TempDf[TempDf$country==i,"welfare"]
  TempRes <- TempWel %*% Weights
  Results2[i] <- TempRes
  }
 OssaTable$WorldWelfare <- Results2


 Results3 <- vector(length=7)
 for(i in 1:7) {
  TempWel <- Out[Out$country==i,"welfare"]
  Results3[i] <- median(TempWel)
  }
 OssaTable$ElseMedian <- Results3

 Results4 <- vector(length=7)
 for(i in 1:7) {
  TempWel <- Out[Out$country==i,"welfare"]
  Results4[i] <- mean(TempWel)
```

```
  }
 OssaTable$ElseSimpAvg <- Results4

 # Add to Databse
 OssaTable$Weights <- Weights
 ShockIndex <- (1:length(Shocks))[Shocks==Shock]
 OssaTables[[ShockIndex]] <- OssaTable

}

# manipulate data
GlobalWelfare <- cast( melt(OssaTables) , L1 + country ~ variable)

# merge in the raw shocks
names(GlobalWelfare)[1] <- "shock"
GlobalWelfare  <- merge( GlobalWelfare, data.frame( "rawshock"=Shocks,"shock"=(1:length(Shocks)) ) )

# lose some columns
GlobalWelfare <- GlobalWelfare[,c(10,2,9,3,5,6)]

write.csv( GlobalWelfare, file=Pst("OssaTable",Suffix), row.names = FALSE )
```

```{r TariffAnalysis}

AllTariffs <- melt( readMat('alltariffs.mat') )[,-6]
names(AllTariffs) <- c("exporter_taxed","importer_taxing","industry","shock","value")

ShockDf <- data.frame(shock=1:length(Shocks),rawshock=Shocks)
mAllTariffs <- merge(AllTariffs,ShockDf)

write.csv(mAllTariffs,Pst("AllTariffs",Suffix))
```

End of Document