

This document provide a *Python script* that gives an example of how to read one O3 CDR orbit, filter good orbits, do some unit conversion and plot profiles and maps.
It computes the **total column ozone**: **coltot**

The example uses the orbit file: `W_XX-EUMETSAT-Darmstadt,HYPERSPECT+SOUNDING,METOPB+O3+IASI_C_EUMP_20130221084159Z_20130221102055Z_eps_r_l2_0100.nc`

You need to also have a text file called `Sa_O3.CSV`

This csv file provides the a-priori covariance matrix. It can be created by copying the provided values at the end of this document.

Python script

```
#!/usr/bin/env python
# coding: utf-8

import sys
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from mpl_toolkits.axes_grid1 import make_axes_locatable
import matplotlib.axes as maxes

import cartopy as cart

import cartopy.crs as ccrs
import cartopy.feature as cfeature
from mpl_toolkits.axes_grid1 import make_axes_locatable

import csv
import math

import xarray as xr
import glob

#
# This notebook was created by Marie Doutriaux-Boucher:
# marie.doutriauxboucher@eumetsat.int
#

def mmm_print(var, texte):
    print(texte+' no filter:', np.min(var), np.max(var), np.mean(var), np.median(var), np.std(var))
    return

def convert_ozone(molecules_per_cm2):
    """
    Convert ozone column amount from molecules/cm² to kg/m² and Dobson Units (DU).

    Parameters:
    molecules_per_cm2 (array-like): Ozone column amount in molecules per cm².

    Returns:
    dict: A dictionary with values in kg/m² and DU as numpy arrays.
    """
    # Constants
    AVOGADRO = 6.022e23 # molecules/mol
    MOLAR_MASS_O3 = 0.048 # kg/mol - 48g/mol
    CM2_TO_M2 = 1e4 # 1 cm² = 10⁻⁴ m²
    DU_CONVERSION = 2.687e16 # molecules/cm² per 1 DU

    # Ensure input is a numpy array
    molecules_per_cm2 = np.asarray(molecules_per_cm2)
```

```

# Conversion to kg/m²
kg_per_m2 = (molecules_per_cm2 / CM2_TO_M2) * (MOLAR_MASS_O3 / AVOGADRO)

# Conversion to DU
du_value = molecules_per_cm2 / DU_CONVERSION

#return {"kg/m²": kg_per_m2, "DU": du_value}
return kg_per_m2, du_value

proj_dic = {'Robin':ccrs.Robinson()}

def create_map(plot_file, lat, lon , value, projection='Robin', vminv=0, vmaxv=0.5,
               c_colour='YlOrRd', title_lab = '',
               title_str=None,psize=0.5,colorbar_labsize=25,dpi=100):

    # Marie Doutriaux-Boucher - 28 March 2025

    proj = proj_dic[projection] # decide the proj see dict before the routine

    fig = plt.figure(figsize=(10,5))
    ax = fig.add_subplot(1, 1, 1, projection=proj)
    ax.coastlines(linewidth=3,color='0.05')
    ax.gridlines(linestyle='--', color='black')

    ax.set_global()

    if title_str:
        plt.title(title_str)

    myticks = np.arange(vminv,vmaxv+0.0001,(vmaxv-vminv)/10)

    mymap = c_colour

    # do the plot

    im = ax.scatter(lon[:,], lat[:,], c=value[:,],cmap=mymap,transform=ccrs.PlateCarree(),
                    vmin=vminv,vmax=vmaxv,s=psize,alpha=1)

    # makes the colorbar of the same size as the map
    divider = make_axes_locatable(ax)
    cax = divider.append_axes("right", size="3%", pad=0.1, axes_class=matplotlib.axes.Axes)

    # plot the color_bar
    cbar = fig.colorbar(im,ax=cax,ticks=myticks,label=title_lab,cax = cax)

    cbar.set_label(label=title_lab,size=colorbar_labsize,weight='normal') # change size of the
    colorbar title

    # see
    https://matplotlib.org/stable/api/text_api.html for font weight
    im.figure.axes[1].tick_params(axis="y", labelsize=colorbar_labsize) #change the size of
    colorbar ticks label

    # save the plot
    plt.savefig(plot_file, dpi=dpi, bbox_inches='tight')
    plt.show()
    plt.close()

# ### Read a priori variance-covariance matrix, provided in the section 6.1 of the CO PUM
# !!!! need to have Sa_O3.csv file

Sa_O3_init = pd.read_csv('Sa_O3.csv',header=None,index_col=False,sep=';')
Sa_O3 = Sa_O3_init.values
del(Sa_O3_init)

```

```
print(len(Sa_O3))
print(Sa_O3.shape)
```

```
# ### Provide one single orbit of the L2 O3 CDR netcdf file to read
```

```
# choose the file
# if the file is where you run this script
year = '2013'
month = '01'
day = '01'
sat = 'M01'

#path_to_co_cdr_file = sat+'/netcdf/'+year+'/'+month+'/'+day+'/'
path_to_o3_cdr_file = './'
orbit_o3 = 'W XX-EUMETSAT-
Darmstadt,HYPERSPECT+SOUNDING,METOPB+O3+IASI_C_EUMP_20130221084159Z_20130221102055Z_eps_r_l2_0
100.nc'

netcdf_inputfile = path_to_o3_cdr_file+orbit_o3

print(netcdf_inputfile)

# ## reading with xarray
xrfile = xr.open_dataset(netcdf_inputfile)

# display the variables
print(xrfile.data_vars)

variables =
['forli_layer_heights_o3','lat','lon','o3_npca','o3_qflag','o3_cp_o3_a','o3_cp_air','o3_x_o3',
'o3_nfitlayers','o3_npca','o3_h_eigenvectors','o3_h_eigenvalues']

xrfile[variables]

along_track = xrfile.sizes['along_track']
across_track = xrfile.sizes['across_track']
nl_o3 = xrfile.sizes['nl_o3']
o3_levels = xrfile['forli_layer_heights_o3'].values # O3 levels in m

print(o3_levels)

o3_qflag = xrfile['o3_qflag']
mmm_print(o3_qflag,'o3_qflag')

print(np.min(o3_qflag.values),np.max(o3_qflag.values))
```

```
# ### do some tests to keep only valid profiles
# NEEDED to check later valid profiles - test 11
```

```
tmp = xrfile['o3_h_eigenvalues']
npca_values = xrfile['o3_npca'].values

# Replace NaN values with 0 and convert to integer
npca_values[np.isnan(npca_values)] = 0
npca_values = npca_values.astype(int)

for i in range(along_track):
    for j in range(across_track):
        ind = npca_values[i, j]
        tmp[i, j, ind:] = 0
```

```

test1 = xrfile['o3_nfitlayers']>-1
test2 = xrfile['o3_x_o3'].std(axis=2) > 0 # check that co_x_co is not constant
# (e.g. if standard dev >0)
test3 = (xrfile['o3_x_o3']==0).sum(axis=2) == 0 # check that co_x_co profile is not
# full of 0 (sum all lev with value 0)
test4 = (xrfile['o3_x_o3']>9.96e36).sum(axis=2) == 0 # check that co_x_co profile do not
# contain value > 9.96e3
test5 = (xrfile['o3_x_o3']<1e-5).sum(axis=2) == 0 # check that profile do not contain
# value <1E-5
test6 = ((xrfile['o3_x_o3']>650000) &
          (xrfile['o3_x_o3']<660000)).sum(axis=2) == 0 # check that co_x_co profile has no
# values >650000 and <660000
test7 = (xrfile['o3_cp_o3_a']==0).sum(axis=2) == 0 # check that the prior, co_xp_co_a,
# profile is not full of 0
test8 = xrfile['o3_x_o3'].isnull().sum(axis=2) == 0 # check that co_c_co has no nan
test9 = (xrfile['lat']<90) & (xrfile['lat']>-90) # check that lat has valid values
test10 = xrfile['o3_qflag']>-1 # check the flag > -1
test11 = xrfile['o3_h_eigenvalues'].sum(axis=2) == xrfile['o3_npca'] # check that co_npca
# (num of vector
# describing the matrix)
# is equal to the num of
# eigen value !!! need
# the prior rearrangment
test12 = (xrfile['o3_cp_air']==0).sum(axis=2) == 0 # check that air profil, o3_cp_air,
# profile is not full of 0
keep = xr.where( test1 & test2 & test3 & test4 & test5 & test6 & test7 & test8 & test9 &
test10 & test11 & test12, True, False)
print(keep.sum())

```

```

# ### keep some variables for the 'valid' profiles and compute the total column ozone coltot

```

```

keep_indices = ~xrfile['lat'].where(keep).isnull()

lat      = xrfile['lat'].values[keep_indices]
lon      = xrfile['lon'].values[keep_indices]
scalfac  = xrfile['o3_x_o3'].values[keep_indices]
prior    = xrfile['o3_cp_o3_a'].values[keep_indices]
air      = xrfile['o3_cp_air'].values[keep_indices]
fit      = xrfile['o3_cp_o3_a'].values[keep_indices]*xrfile['o3_x_o3'].values[keep_indices]
coltot   = np.nansum(fit,axis=1)
nfitlay  = xrfile['o3_nfitlayers'].values[keep_indices]
eve      = xrfile['o3_h_eigenvectors'].values[keep_indices]
eva      = xrfile['o3_h_eigenvalues'].values[keep_indices]
npca     = xrfile['o3_npca'].values[keep_indices]
surfp    = xrfile['surface_pressure'].values[keep_indices]
gqf      = xrfile['o3_qflag'].values[keep_indices]

```

```

print('lat shape:',lat.shape)
print('lon shape:',lat.shape)
print('fit shape',fit.shape)
print('air.shape',air.shape)
print('scalfac.shape',scalfac.shape)
print('coltot shape',coltot.shape)
print('eve.shape',eve.shape)
print('scalfac shape:',scalfac.shape)
print('co_nfitlayers.shape ',nfitlay.shape)
print('npca.shape',npca.shape)
print('prior.shape',prior.shape)
print('eva.shape',eva.shape)
print('surfp.shape',surfp.shape)

```

```

# # reconstruct matrices + error estimation -

```

```

ERROR_PROF  = np.tile(np.nan,(len(lat),nl_o3))
ERROR_COLTOT = np.tile(np.nan,(len(lat)))

AVK          = np.tile(np.nan,(len(lat),nl_o3,nl_o3))

```

```

dofs          = np.tile(np.nan, (len(lat)))

for i in range(lat.size):

    # matrices reconstruction + errors estimation
    tmp_eve     = eve.take(i,axis=0)
    tmp_eve2    = tmp_eve.take(range(0,int(nfitlay[i])*int(npca[i]))))
    transp_nu   = np.reshape(tmp_eve2, (int(npca[i]),int(nfitlay[i])))
    del(tmp_eve)
    del(tmp_eve2)

    nu          = np.transpose(transp_nu)
    tmp_eva     = eva.take(i,axis=0)
    tmp_eva2    = tmp_eva.take(range(0,int(npca[i])))

    dia_lambda  = np.diag(tmp_eva2)
    del(tmp_eva)
    del(tmp_eva2)

    H1 = nu.dot(dia_lambda)
    H  = H1.dot(transp_nu)
    del(H1)

    Sa = Sa_O3.take(range(41-int(nfitlay[i]),41),axis=1)
    Sa = Sa.take(range(41-int(nfitlay[i]),41),axis=0)

    inv_Sa = np.linalg.inv(Sa)
    S       = np.linalg.inv(H+inv_Sa)

    ERROR_PROF1 = np.divide(np.sqrt(np.diag(S)),scalfac[i,range(41-int(nfitlay[i]),41)])
    ERROR_PROF[i,range(41-int(nfitlay[i]),41)] = ERROR_PROF1

    S_CP1 = np.diag(prior[i,range(41-int(nfitlay[i]),41)]) .dot(S)
    S_CP  = S_CP1.dot(np.diag(prior[i,range(41-int(nfitlay[i]),41)]))
    del(S_CP1)

    ERROR_COLTOT[i]=np.sqrt(np.sum(np.sum(S_CP)))/np.sum(fit[i,range(41-int(nfitlay[i]),19)])
    del(S_CP)

    AVK_SF = S.dot(H)
    del(H)
    del(S)

    AVK_CP1 = np.diag(prior[i,range(41-int(nfitlay[i]),41)]) .dot(AVK_SF)
    AVK_CP  = AVK_CP1.dot(np.linalg.inv(np.diag(prior[i,range(41-int(nfitlay[i]),41)])))
    del(AVK_CP1)

    dofs[i] = np.sum(np.diag(AVK_CP))

    AVK[i,41-int(nfitlay[i]):,41-int(nfitlay[i]):] = AVK_CP

    del(AVK_CP)

print(np.max(coltot),np.mean(coltot))

```

```

#
# # plot the map of total column O3
#

dirout = './'

mini   = 1.E17 #
maxi   = 2.E19 #

fileout = dirout+'test_O3_valid_marie_'+sat+'_'+day+'_'+month+'_'+year+'_1_orb.png'
colorp  = 'jet'
proj    = 'Robin'
title_str=''

```

```

title_str= sat+' '+day+' '+month+' '+year

create_map(fileout, lat, lon, coltot, projection = proj,
           vminv = mini,vmaxv=maxi, c_colour=colorp, title_lab='O3 totcol
[molecules/cm$^2$]', title_str=title_str,
           psize=5,colorbar_labsz=15)

coltot_kg_per_m2,coltotDU = convert_ozone(coltot)

mmm_print(coltot_kg_per_m2,'coltot_kg_per_m2')

dirout = './'

mini = 0 #
maxi = 550 #

fileout = dirout+'test_O3_valid_marie_'+sat+'_'+day+'_'+month+'_'+year+'_1_orb.png'
colorp = 'jet'
proj = 'Robin'
title_str=''
title_str= sat+' '+day+' '+month+' '+year

create_map(fileout, lat, lon,coltotDU, projection = proj,
           vminv = mini,vmaxv=maxi, c_colour=colorp, title_lab='O3 totcol [DU]',
           title_str=title_str,
           psize=5,colorbar_labsz=15)

```

plot one of the good profile

```

obs = 1 #choose the profile
print(fit[1,:])

```

in molecule/cm2/layer of 1 km (fit)

```

# plot one profile at one location
print(o3_levels)
fig, ax1 = plt.subplots(figsize=(7,10))
fontsize=14

xmin = 1.E15
xmax = 12.5E17

ax1.set_ylabel('Altitude (km)',fontsize=22)
#ax1.set_xlabel('')
ax1.set_xlim(xmin,xmax)
ax1.set_ylim(0,20)
plt.xticks(fontsize=fontsize)
ax1.set_yticks([0,2,4,6,8,10,12,14,16,18])
ax1.set_yticklabels([0,2,4,6,8,10,12,14,16,18],fontsize=fontsize)
ax1.set_xlabel('O3 (molecules/cm$^2$/layer of 1 km)',fontsize = fontsize+5)

ax1.plot(fit[obs,:-1],(o3_levels[:-1]+500)/1000.,color='black',marker='o')

for i in range(nl_o3):
    #i=i+1
    #print('i=',i,' pp[i]=' ,pp[i])
    ax1.plot([xmin,xmax],[o3_levels[i]/1000.,o3_levels[i]/1000.],color='grey',linewidth=0.2)

plt.title('lat: '+str(round(lat[obs],4))+' , lon: '+str(round(lon[obs],4)))

plt.savefig('Test_O3_prof_41lev.png',dpi=100,bbx_inches='tight')

```

in ppb : fit/air*1E9

```

# plot one profile at one location

```

```

print('o3 levels in m:',o3_levels)

coltot_ppb = np.sum(fit[obs,:-1])/np.sum(air[obs,:-1])*1e9 # convert into ppb
fig, ax1 = plt.subplots(figsize=(7,10))
fontsize=14

xmin = 20
xmax = 6000

ax1.set_ylabel('Altitude (km)',fontsize=22)
#ax1.set_xlabel('')
ax1.set_xlim(xmin,xmax)
ax1.set_ylim(0,20)
plt.xticks(fontsize=fontsize)
ax1.set_yticks([0,2,4,6,8,10,12,14,16,18])
ax1.set_yticklabels([0,2,4,6,8,10,12,14,16,18],fontsize=fontsize)
ax1.set_xlabel('O3 (ppb)',fontsize = fontsize+5)

ax1.plot(fit[obs,:-1]/air[obs,:-1]*1e9, (o3_levels[:-1]+500)/1000.,color='black',marker="o")

plt.title('lat:'+str(round(lat[obs],4))+', lon:'+str(round(lon[obs],4))+',
coltot='+str(round(coltot_ppb,3))+ ' ppb')

for i in range(nl_o3):
    ax1.plot([xmin,xmax],[o3_levels[i]/1000.,o3_levels[i]/1000.],color='grey',linewidth=0.2)

plt.savefig('Test_O3_prof_41lev_ppb.png',dpi=100,bbox_inches='tight')

```

```

# # How to save/prints? some example

```

```

obs = 2
print('obs=',obs,'lat = ',lat[obs],' lon=',lon[obs],'surfp = ',surfp[obs])
print('-----')
mmm_print(fit[obs,:-1],'fit = profile in molecule/cm2/layer of 1km: ')
mmm_print(fit[obs,:-1]/air[obs,:-1]*1e9,'fit/air*1e9 = profile in ppb')
print()

print('coltot in ppm from sum of each layers:sum fit/air in ppb',sum(fit[obs,:-1])/sum(air[obs,:-1])*1e9)

print('coltot in molecule/cm2:',coltot[obs])
print()
print('fit',fit[obs,:])

print('fit in ppm per layer of 1km:',fit[obs,:-1]/air[obs,:-1]*1e9)

```

```

# # save the valid profiles in a dataframe

```

```

if 'df' in globals():
    del df

# create the dataframe
df = pd.DataFrame(lat,columns=['lat'])
# add other columns
df['lon'] = lon
df['surfp'] = surfp[:]
df['gqf'] = gqf[:]
df['nfitlay'] = nfitlay[:]
df['fit'] = list(fit[:,:])
df['fit_ppb'] = list(fit[:,:]/air[:,:]*1e9)
df['ERROR_PROF'] = list(ERROR_PROF[:,:])
df['coltot'] = coltot[:]
df['ERROR_COLTOT'] = ERROR_COLTOT[:]
df['dofs'] = dofs

```

```
df
```

```
df.to_csv('Good_profiles_O3.csv', index=False)
```

```
# # read the newly created dataframe
```

```
test_df=pd.read_csv('Good_profiles.csv',index_col=False)
```

```
print(test_df['fit_ppb'][0])
```

```
print(lat[0])
```


This part shows the content of the file: **Sa_O3.csv**

You need to copy the 41 lines and save them in a file named `Sa_O3.csv`

Each of the 41 lines should contain 41 elements separated by a semicolon (;)

```
9.15713540e-02;8.04418740e-02;6.78846030e-02;5.67218210e-02;4.90903010e-
02;4.59661230e-02;4.44182540e-02;4.25864680e-02;3.80634580e-02;3.12143360e-
02;3.19702880e-02;4.03399390e-02;4.66916150e-02;4.86645000e-02;4.53687900e-
02;4.08420110e-02;3.67740940e-02;3.14132290e-02;2.62424640e-02;2.20382100e-
02;1.97281800e-02;1.68621120e-02;1.32037430e-02;9.89389620e-03;6.19372390e-
03;2.31088070e-03;-1.11853050e-03;-3.99407330e-03;-6.18178530e-03;-
7.79284340e-03;-8.50120830e-03;-8.42080950e-03;-7.65892220e-03;-
6.40761030e-03;-5.12864270e-03;-3.89231390e-03;-2.86429520e-03;-
2.06116170e-03;-1.55041830e-03;-1.10416420e-03;3.71254550e-03
8.04418740e-02;7.76543230e-02;7.00595030e-02;6.05290770e-02;5.31958610e-
02;4.94773370e-02;4.78001760e-02;4.73489130e-02;4.71024380e-02;4.58354520e-
02;4.99037380e-02;5.84910470e-02;6.29189630e-02;6.23433730e-02;5.70820740e-
02;5.13113960e-02;4.60103530e-02;4.00499180e-02;3.40727100e-02;2.88788310e-
02;2.53378320e-02;2.13613780e-02;1.67727040e-02;1.26052820e-02;8.11165660e-
03;3.52018510e-03;-5.49730700e-04;-4.01318130e-03;-6.66614070e-03;-
8.58022680e-03;-9.45628510e-03;-9.36479230e-03;-8.42399950e-03;-
6.90247320e-03;-5.31118480e-03;-3.78424280e-03;-2.48403250e-03;-
1.41684620e-03;-6.89173660e-04;-3.38528910e-05;3.41629837e-03
6.78846030e-02;7.00595030e-02;7.35065170e-02;6.80896000e-02;6.23066250e-
02;5.85910260e-02;5.61482120e-02;5.49627410e-02;5.26283040e-02;4.62145860e-
02;4.92009280e-02;5.61436080e-02;5.73678080e-02;5.32749620e-02;4.55835410e-
02;3.82815570e-02;3.25434170e-02;2.77241050e-02;2.42046230e-02;2.20551180e-
02;2.09369710e-02;1.91975670e-02;1.70254840e-02;1.49327950e-02;1.23582670e-
02;9.56846140e-03;7.04198310e-03;4.82250510e-03;3.18388910e-03;2.00380420e-
03;1.62262700e-03;1.95402530e-03;2.86783770e-03;4.04730860e-03;5.15381810e-
03;6.05647540e-03;6.71376760e-03;7.02889770e-03;7.05130740e-03;6.91493550e-
03;5.17981497e-03
5.67218210e-02;6.05290770e-02;6.80896000e-02;6.73879200e-02;6.32656260e-
02;5.99958150e-02;5.69888020e-02;5.48699890e-02;5.00777070e-02;3.95555080e-
02;3.96830510e-02;4.33271840e-02;4.16723490e-02;3.56469590e-02;2.73970120e-
02;2.00437520e-02;1.46477960e-02;1.10955140e-02;1.00063900e-02;1.08217300e-
02;1.20240300e-02;1.23608050e-02;1.24181230e-02;1.24588420e-02;1.20242910e-
02;1.13213740e-02;1.06818450e-02;1.00960800e-02;9.83008180e-03;9.65112600e-
03;9.89043910e-03;1.04479790e-02;1.12523070e-02;1.19576800e-02;1.24151060e-
02;1.25416940e-02;1.24147980e-02;1.18762980e-02;1.10803550e-02;1.00762980e-
02;5.01996838e-03
4.90903010e-02;5.31958610e-02;6.23066250e-02;6.32656260e-02;6.11232630e-
02;5.86865820e-02;5.61088490e-02;5.39941470e-02;4.86487390e-02;3.73337070e-
02;3.62881930e-02;3.79722160e-02;3.45083520e-02;2.73192960e-02;1.88509270e-
02;1.17673060e-02;6.85727900e-03;4.04176710e-03;3.97065070e-03;5.89406100e-
03;7.88969870e-03;8.95260940e-03;9.81819770e-03;1.06634860e-02;1.11733880e-
02;1.15333210e-02;1.19167670e-02;1.22625070e-02;1.27737730e-02;1.31646710e-
02;1.37687800e-02;1.44822270e-02;1.52040130e-02;1.56237740e-02;1.57230480e-
02;1.54052920e-02;1.48232520e-02;1.37921340e-02;1.24874820e-02;1.09206240e-
02;4.01331378e-03
4.59661230e-02;4.94773370e-02;5.85910260e-02;5.99958150e-02;5.86865820e-
02;5.83260450e-02;5.66393580e-02;5.56214060e-02;5.21695230e-02;4.34509060e-
02;4.27301830e-02;4.22125230e-02;3.68527940e-02;2.86333230e-02;2.00104810e-
02;1.28950880e-02;7.80575620e-03;4.69123380e-03;4.38166250e-03;5.99221560e-
03;7.52875560e-03;8.22803800e-03;8.81618360e-03;9.43880630e-03;9.85694540e-
03;1.02733310e-02;1.08047340e-02;1.13758420e-02;1.21594080e-02;1.27947430e-
```

02;1.36009500e-02;1.44305220e-02;1.51562160e-02;1.55100830e-02;1.55000460e-02;1.50409740e-02;1.42759200e-02;1.30606990e-02;1.15020730e-02;9.66638330e-03;2.09588537e-03
4.44182540e-02;4.78001760e-02;5.61482120e-02;5.69888020e-02;5.61088490e-02;5.66393580e-02;5.67390200e-02;5.76591980e-02;5.85346000e-02;5.56907410e-02;5.77986510e-02;5.66333350e-02;5.02184450e-02;4.15522140e-02;3.30072870e-02;2.59795680e-02;2.03698820e-02;1.61839460e-02;1.40893270e-02;1.35498790e-02;1.30226920e-02;1.19906380e-02;1.09198410e-02;1.00223180e-02;9.09953960e-03;8.39620930e-03;8.01880210e-03;7.88587690e-03;8.15923330e-03;8.49542510e-03;9.17248980e-03;9.99445890e-03;1.08018850e-02;1.13775450e-02;1.16793140e-02;1.15500350e-02;1.10781540e-02;1.01975480e-02;8.87087600e-03;7.25725550e-03;4.68012767e-04
4.25864680e-02;4.73489130e-02;5.49627410e-02;5.48699890e-02;5.39941470e-02;5.56214060e-02;5.76591980e-02;6.47653480e-02;7.84724760e-02;9.32100320e-02;1.04325390e-01;1.02589290e-01;9.39638790e-02;8.47827820e-02;7.64160110e-02;6.90259660e-02;6.11463440e-02;5.24921350e-02;4.45936180e-02;3.73594240e-02;3.07462780e-02;2.44803370e-02;1.84693690e-02;1.30325630e-02;7.96897730e-03;3.56117550e-03;9.56397900e-06;-2.70652150e-03;-4.31551240e-03;-
5.16294350e-03;-5.09486650e-03;-4.39585710e-03;-3.27759400e-03;-
1.93553110e-03;-6.05837780e-04;4.27512070e-04;1.05355580e-03;1.43039060e-03;1.11864690e-03;5.60643830e-04;-2.76679346e-03
3.80634580e-02;4.71024380e-02;5.26283040e-02;5.00777070e-02;4.86487390e-02;5.21695230e-02;5.85346000e-02;7.84724760e-02;1.25579140e-01;1.85592320e-01;2.19642020e-01;2.16312060e-01;2.02254370e-01;1.92148520e-01;1.84001230e-01;1.75271990e-01;1.61314080e-01;1.41626620e-01;1.19053840e-01;9.53461190e-02;7.37812950e-02;5.45736610e-02;3.64349720e-02;1.98755900e-02;4.82421350e-03;-8.46329830e-03;-1.96000690e-02;-2.84591520e-02;-3.45243650e-02;-
3.81945080e-02;-3.96293600e-02;-3.92232580e-02;-3.73222460e-02;-
3.41123160e-02;-3.02759950e-02;-2.63884050e-02;-2.30340310e-02;-
1.95120860e-02;-1.72809990e-02;-1.52587280e-02;-1.07124776e-02
3.12143360e-02;4.58354520e-02;4.62145860e-02;3.95555080e-02;3.73337070e-02;4.34509060e-02;5.56907410e-02;9.32100320e-02;1.85592320e-01;3.15871210e-01;3.85078690e-01;3.80017330e-01;3.58981180e-01;3.48486990e-01;3.41871870e-01;3.32925420e-01;3.11563150e-01;2.75835820e-01;2.31470510e-01;1.82565240e-01;1.37930980e-01;9.89091990e-02;6.21964490e-02;2.85929900e-02;-
1.76865400e-03;-2.87606390e-02;-5.15747450e-02;-6.98306880e-02;-
8.26038470e-02;-9.04597760e-02;-9.41107350e-02;-9.41034660e-02;-
9.10059510e-02;-8.49125980e-02;-7.72204280e-02;-6.89423180e-02;-
6.13698170e-02;-5.29457550e-02;-4.67017940e-02;-4.05531580e-02;-
2.29238276e-02
3.19702880e-02;4.99037380e-02;4.92009280e-02;3.96830510e-02;3.62881930e-02;4.27301830e-02;5.77986510e-02;1.04325390e-01;2.19642020e-01;3.85078690e-01;4.85941840e-01;4.93049980e-01;4.74602360e-01;4.65656820e-01;4.59061590e-01;4.48446430e-01;4.21385630e-01;3.74387590e-01;3.14368230e-01;2.47452920e-01;1.86703330e-01;1.33666540e-01;8.39154650e-02;3.82609400e-02;-
3.10807080e-03;-3.99224800e-02;-7.11779520e-02;-9.64015750e-02;-
1.14382170e-01;-1.25508160e-01;-1.30937500e-01;-1.31273500e-01;-
1.27233990e-01;-1.18963580e-01;-1.08389630e-01;-9.68248460e-02;-
8.61107340e-02;-7.41277180e-02;-6.48370350e-02;-5.55341740e-02;-
2.70210087e-02
4.03399390e-02;5.84910470e-02;5.61436080e-02;4.33271840e-02;3.79722160e-02;4.22125230e-02;5.66333350e-02;1.02589290e-01;2.16312060e-01;3.80017330e-01;4.93049980e-01;5.22209700e-01;5.17815390e-01;5.15000660e-01;5.08964670e-01;4.96834580e-01;4.67580200e-01;4.16748440e-01;3.50425490e-01;2.76131120e-01;2.09484760e-01;1.51207980e-01;9.65713860e-02;4.63809370e-02;5.93905940e-04;-4.03397560e-02;-7.53364040e-02;-1.03927430e-01;-1.24866710e-01;-
1.38000710e-01;-1.44678780e-01;-1.45475590e-01;-1.41183150e-01;-

1.31989310e-01;-1.20187480e-01;-1.07195110e-01;-9.50340250e-02;-
8.15191680e-02;-7.06370380e-02;-5.95539630e-02;-2.21019300e-02
4.66916150e-02;6.29189630e-02;5.73678080e-02;4.16723490e-02;3.45083520e-
02;3.68527940e-02;5.02184450e-02;9.39638790e-02;2.02254370e-01;3.58981180e-
01;4.74602360e-01;5.17815390e-01;5.28158820e-01;5.34274400e-01;5.31949490e-
01;5.20027510e-01;4.89723740e-01;4.37193400e-01;3.67565230e-01;2.89052580e-
01;2.19458920e-01;1.58761130e-01;1.01895930e-01;4.97698040e-02;2.09548040e-
03;-4.07774990e-02;-7.75714990e-02;-1.07797870e-01;-1.30337740e-01;-
1.44615290e-01;-1.52125810e-01;-1.53385820e-01;-1.49179480e-01;-
1.39691680e-01;-1.27408800e-01;-1.13799120e-01;-1.00968630e-01;-
8.66769670e-02;-7.48487060e-02;-6.25845430e-02;-1.67392046e-02
4.86645000e-02;6.23433730e-02;5.32749620e-02;3.56469590e-02;2.73192960e-
02;2.86333230e-02;4.15522140e-02;8.47827820e-02;1.92148520e-01;3.48486990e-
01;4.65656820e-01;5.15000660e-01;5.34274400e-01;5.50586410e-01;5.55001870e-
01;5.45995790e-01;5.15330300e-01;4.60188850e-01;3.86286110e-01;3.02569810e-
01;2.28860360e-01;1.64867700e-01;1.05199030e-01;5.06256020e-02;8.51570520e-
04;-4.41780100e-02;-8.28518240e-02;-1.14673730e-01;-1.38650800e-01;-
1.53990260e-01;-1.62308200e-01;-1.64082040e-01;-1.60073740e-01;-
1.50351040e-01;-1.37622700e-01;-1.23400130e-01;-1.09861610e-01;-
9.46554480e-02;-8.17857940e-02;-6.82387580e-02;-1.30619144e-02
4.53687900e-02;5.70820740e-02;4.55835410e-02;2.73970120e-02;1.88509270e-
02;2.00104810e-02;3.30072870e-02;7.64160110e-02;1.84001230e-01;3.41871870e-
01;4.59061590e-01;5.08964670e-01;5.31949490e-01;5.55001870e-01;5.67470420e-
01;5.63459810e-01;5.34458230e-01;4.78228180e-01;4.01197620e-01;3.13438640e-
01;2.36162000e-01;1.69276850e-01;1.07114800e-01;5.04027910e-02;-
1.17264330e-03;-4.80052990e-02;-8.82043560e-02;-1.21314460e-01;-
1.46414550e-01;-1.62647080e-01;-1.71625890e-01;-1.73828990e-01;-
1.70049860e-01;-1.60122780e-01;-1.47020220e-01;-1.32260580e-01;-
1.18070230e-01;-1.02051060e-01;-8.82558410e-02;-7.36124940e-02;-
1.07139874e-02
4.08420110e-02;5.13113960e-02;3.82815570e-02;2.00437520e-02;1.17673060e-
02;1.28950880e-02;2.59795680e-02;6.90259660e-02;1.75271990e-01;3.32925420e-
01;4.48446430e-01;4.96834580e-01;5.20027510e-01;5.45995790e-01;5.63459810e-
01;5.65954720e-01;5.41098310e-01;4.86101180e-01;4.08602860e-01;3.19240250e-
01;2.40227510e-01;1.71678440e-01;1.08003250e-01;4.99039430e-02;-
2.87230360e-03;-5.08342360e-02;-9.19754270e-02;-1.25943100e-01;-
1.51836470e-01;-1.68772030e-01;-1.78240810e-01;-1.80745110e-01;-
1.77146080e-01;-1.66990560e-01;-1.53543870e-01;-1.38338230e-01;-
1.23551680e-01;-1.06877320e-01;-9.24021210e-02;-7.70454290e-02;-
9.12976283e-03
3.67740940e-02;4.60103530e-02;3.25434170e-02;1.46477960e-02;6.85727900e-
03;7.80575620e-03;2.03698820e-02;6.11463440e-02;1.61314080e-01;3.11563150e-
01;4.21385630e-01;4.67580200e-01;4.89723740e-01;5.15330300e-01;5.34458230e-
01;5.41098310e-01;5.21797720e-01;4.71065780e-01;3.97243660e-01;3.10966470e-
01;2.34336850e-01;1.67624820e-01;1.05455440e-01;4.85926850e-02;-
3.16409170e-03;-5.01578180e-02;-9.04995510e-02;-1.23898710e-01;-
1.49448520e-01;-1.66301450e-01;-1.75740190e-01;-1.78283370e-01;-
1.74837480e-01;-1.64802460e-01;-1.51533140e-01;-1.36513950e-01;-
1.21804390e-01;-1.05297520e-01;-9.09060160e-02;-7.56910460e-02;-
7.89632209e-03
3.14132290e-02;4.00499180e-02;2.77241050e-02;1.10955140e-02;4.04176710e-
03;4.69123380e-03;1.61839460e-02;5.24921350e-02;1.41626620e-01;2.75835820e-
01;3.74387590e-01;4.16748440e-01;4.37193400e-01;4.60188850e-01;4.78228180e-
01;4.86101180e-01;4.71065780e-01;4.29154930e-01;3.63866930e-01;2.86158790e-
01;2.16646040e-01;1.55827160e-01;9.89247860e-02;4.68213230e-02;-
7.95197830e-04;-4.40911620e-02;-8.13197820e-02;-1.12273430e-01;-
1.36100640e-01;-1.51945750e-01;-1.60872980e-01;-1.63356540e-01;-

1.60224210e-01;-1.50897170e-01;-1.38548470e-01;-1.24582920e-01;-
1.10907300e-01;-9.56169360e-02;-8.23066640e-02;-6.83109820e-02;-
6.76772549e-03
2.62424640e-02;3.40727100e-02;2.42046230e-02;1.00063900e-02;3.97065070e-
03;4.38166250e-03;1.40893270e-02;4.45936180e-02;1.19053840e-01;2.31470510e-
01;3.14368230e-01;3.50425490e-01;3.67565230e-01;3.86286110e-01;4.01197620e-
01;4.08602860e-01;3.97243660e-01;3.63866930e-01;3.10486540e-01;2.45577000e-
01;1.87060900e-01;1.35659950e-01;8.73685740e-02;4.30110900e-02;2.24591560e-
03;-3.48714500e-02;-6.68369010e-02;-9.35111110e-02;-1.14114280e-01;-
1.27888120e-01;-1.35654930e-01;-1.37847120e-01;-1.35174910e-01;-
1.27131960e-01;-1.16492170e-01;-1.04473280e-01;-9.27142490e-02;-
7.96258660e-02;-6.82574230e-02;-5.63680070e-02;-5.14296457e-03
2.20382100e-02;2.88788310e-02;2.20551180e-02;1.08217300e-02;5.89406100e-
03;5.99221560e-03;1.35498790e-02;3.73594240e-02;9.53461190e-02;1.82565240e-
01;2.47452920e-01;2.76131120e-01;2.89052580e-01;3.02569810e-01;3.13438640e-
01;3.19240250e-01;3.10966470e-01;2.86158790e-01;2.45577000e-01;1.95895290e-
01;1.50734330e-01;1.10855260e-01;7.31023500e-02;3.82254330e-02;5.92188880e-
03;-2.35482990e-02;-4.89862890e-02;-7.03012100e-02;-8.68156810e-02;-
9.79313900e-02;-1.04207340e-01;-1.06002860e-01;-1.03889590e-01;-
9.74773940e-02;-8.90074540e-02;-7.94528890e-02;-7.01144850e-02;-
5.97982540e-02;-5.08571880e-02;-4.15764860e-02;-2.85893296e-03
1.97281800e-02;2.53378320e-02;2.09369710e-02;1.20240300e-02;7.88969870e-
03;7.52875560e-03;1.30226920e-02;3.07462780e-02;7.37812950e-02;1.37930980e-
01;1.86703330e-01;2.09484760e-01;2.19458920e-01;2.28860360e-01;2.36162000e-
01;2.40227510e-01;2.34336850e-01;2.16646040e-01;1.87060900e-01;1.50734330e-
01;1.17785520e-01;8.84533250e-02;6.03147260e-02;3.40410430e-02;9.37602540e-
03;-1.32427490e-02;-3.28458940e-02;-4.93656870e-02;-6.22321410e-02;-
7.09705870e-02;-7.59180180e-02;-7.73606110e-02;-7.57511280e-02;-
7.08074380e-02;-6.42941900e-02;-5.69613850e-02;-4.98148530e-02;-
4.20097760e-02;-3.52636350e-02;-2.83239190e-02;-4.45332526e-04
1.68621120e-02;2.13613780e-02;1.91975670e-02;1.23608050e-02;8.95260940e-
03;8.22803800e-03;1.19906380e-02;2.44803370e-02;5.45736610e-02;9.89091990e-
02;1.33666540e-01;1.51207980e-01;1.58761130e-01;1.64867700e-01;1.69276850e-
01;1.71678440e-01;1.67624820e-01;1.55827160e-01;1.35659950e-01;1.10855260e-
01;8.84533250e-02;6.84592820e-02;4.88866450e-02;3.02652580e-02;1.23999980e-
02;-4.14300830e-03;-1.85706580e-02;-3.08276080e-02;-4.04498810e-02;-
4.70602870e-02;-5.08249220e-02;-5.19688510e-02;-5.08453570e-02;-
4.72512370e-02;-4.25170730e-02;-3.71925950e-02;-3.20232140e-02;-
2.64632850e-02;-2.16525860e-02;-1.67489360e-02;1.82265152e-03
1.32037430e-02;1.67727040e-02;1.70254840e-02;1.24181230e-02;9.81819770e-
03;8.81618360e-03;1.09198410e-02;1.84693690e-02;3.64349720e-02;6.21964490e-
02;8.39154650e-02;9.65713860e-02;1.01895930e-01;1.05199030e-01;1.07114800e-
01;1.08003250e-01;1.05455440e-01;9.89247860e-02;8.73685740e-02;7.31023500e-
02;6.03147260e-02;4.88866450e-02;3.74408170e-02;2.61238620e-02;1.48226830e-
02;4.14990400e-03;-5.26735500e-03;-1.33922670e-02;-1.98705160e-02;-
2.44112730e-02;-2.70282870e-02;-2.78865540e-02;-2.72455700e-02;-
2.49716340e-02;-2.19720650e-02;-1.86006040e-02;-1.53409130e-02;-
1.19200510e-02;-8.93427200e-03;-5.93385850e-03;4.09821215e-03
9.89389620e-03;1.26052820e-02;1.49327950e-02;1.24588420e-02;1.06634860e-
02;9.43880630e-03;1.00223180e-02;1.30325630e-02;1.98755900e-02;2.85929900e-
02;3.82609400e-02;4.63809370e-02;4.97698040e-02;5.06256020e-02;5.04027910e-
02;4.99039430e-02;4.85926850e-02;4.68213230e-02;4.30110900e-02;3.82254330e-
02;3.40410430e-02;3.02652580e-02;2.61238620e-02;2.15900730e-02;1.64562450e-
02;1.13092370e-02;6.60364180e-03;2.35220320e-03;-1.21408930e-03;-
3.84889750e-03;-5.42865100e-03;-6.05507300e-03;-5.88652190e-03;-
4.86196780e-03;-3.48498680e-03;-1.93557420e-03;-4.56913550e-04;9.92673720e-
04;2.30860050e-03;3.58258320e-03;6.00248266e-03

6.19372390e-03;8.11165660e-03;1.23582670e-02;1.20242910e-02;1.11733880e-02;9.85694540e-03;9.09953960e-03;7.96897730e-03;4.82421350e-03;-
1.76865400e-03;-3.10807080e-03;5.93905940e-04;2.09548040e-03;8.51570520e-04;-1.17264330e-03;-2.87230360e-03;-3.16409170e-03;-7.95197830e-04;2.24591560e-03;5.92188880e-03;9.37602540e-03;1.23999980e-02;1.48226830e-02;1.64562450e-02;1.71025470e-02;1.71949910e-02;1.69908320e-02;1.64686380e-02;1.57258410e-02;1.49520880e-02;1.43919560e-02;1.40050320e-02;1.37251560e-02;1.35557610e-02;1.33858820e-02;1.31914090e-02;1.29664570e-02;1.25525930e-02;1.22743280e-02;1.18969330e-02;7.41624628e-03
2.31088070e-03;3.52018510e-03;9.56846140e-03;1.13213740e-02;1.15333210e-02;1.02733310e-02;8.39620930e-03;3.56117550e-03;-8.46329830e-03;-
2.87606390e-02;-3.99224800e-02;-4.03397560e-02;-4.07774990e-02;-
4.41780100e-02;-4.80052990e-02;-5.08342360e-02;-5.01578180e-02;-
4.40911620e-02;-3.48714500e-02;-2.35482990e-02;-1.32427490e-02;-
4.14300830e-03;4.14990400e-03;1.13092370e-02;1.71949910e-02;2.21766350e-02;2.62078640e-02;2.92267690e-02;3.12086820e-02;3.22469070e-02;3.27009310e-02;3.25894190e-02;3.19234070e-02;3.06573060e-02;2.90441130e-02;2.72093450e-02;2.53703130e-02;2.31793400e-02;2.13508020e-02;1.93503580e-02;8.31022897e-03
1.11853050e-03;-5.49730700e-04;7.04198310e-03;1.06818450e-02;1.19167670e-02;1.08047340e-02;8.01880210e-03;9.56397900e-06;-1.96000690e-02;-
5.15747450e-02;-7.11779520e-02;-7.53364040e-02;-7.75714990e-02;-
8.28518240e-02;-8.82043560e-02;-9.19754270e-02;-9.04995510e-02;-
8.13197820e-02;-6.68369010e-02;-4.89862890e-02;-3.28458940e-02;-
1.85706580e-02;-5.26735500e-03;6.60364180e-03;1.69908320e-02;2.62078640e-02;3.39791650e-02;4.01723470e-02;4.46331820e-02;4.73391670e-02;4.87396140e-02;4.89010270e-02;4.78945850e-02;4.56521770e-02;4.27465500e-02;3.94338630e-02;3.61315090e-02;3.23259360e-02;2.90665430e-02;2.55672120e-02;8.79772472e-03
-3.99407330e-03;-4.01318130e-03;4.82250510e-03;1.00960800e-02;1.22625070e-02;1.13758420e-02;7.88587690e-03;-2.70652150e-03;-2.84591520e-02;-
6.98306880e-02;-9.64015750e-02;-1.03927430e-01;-1.07797870e-01;-
1.14673730e-01;-1.21314460e-01;-1.25943100e-01;-1.23898710e-01;-
1.12273430e-01;-9.35111110e-02;-7.03012100e-02;-4.93656870e-02;-
3.08276080e-02;-1.33922670e-02;2.35220320e-03;1.64686380e-02;2.92267690e-02;4.01723470e-02;4.91391440e-02;5.58237430e-02;6.00605750e-02;6.23603090e-02;6.28192770e-02;6.15484760e-02;5.84666280e-02;5.44343350e-02;4.98200920e-02;4.52101860e-02;3.99595680e-02;3.53963980e-02;3.05331150e-02;8.82674905e-03
-6.18178530e-03;-6.66614070e-03;3.18388910e-03;9.83008180e-03;1.27737730e-02;1.21594080e-02;8.15923330e-03;-4.31551240e-03;-3.45243650e-02;-
8.26038470e-02;-1.14382170e-01;-1.24866710e-01;-1.30337740e-01;-
1.38650800e-01;-1.46414550e-01;-1.51836470e-01;-1.49448520e-01;-
1.36100640e-01;-1.14114280e-01;-8.68156810e-02;-6.22321410e-02;-
4.04498810e-02;-1.98705160e-02;-1.21408930e-03;1.57258410e-02;3.12086820e-02;4.46331820e-02;5.58237430e-02;6.43859850e-02;6.99530860e-02;7.30669390e-02;7.38411730e-02;7.24194510e-02;6.86961320e-02;6.37722790e-02;5.81092630e-02;5.24216370e-02;4.59768080e-02;4.03082210e-02;3.42810450e-02;8.36643346e-03
-7.79284340e-03;-8.58022680e-03;2.00380420e-03;9.65112600e-03;1.31646710e-02;1.27947430e-02;8.49542510e-03;-5.16294350e-03;-3.81945080e-02;-
9.04597760e-02;-1.25508160e-01;-1.38000710e-01;-1.44615290e-01;-
1.53990260e-01;-1.62647080e-01;-1.68772030e-01;-1.66301450e-01;-
1.51945750e-01;-1.27888120e-01;-9.79313900e-02;-7.09705870e-02;-
4.70602870e-02;-2.44112730e-02;-3.84889750e-03;1.49520880e-02;3.22469070e-02;4.73391670e-02;6.00605750e-02;6.99530860e-02;7.65123530e-02;8.02614900e-02;8.13108360e-02;7.98271060e-02;7.56843350e-02;7.01584050e-02;6.37724120e-

02;5.73261140e-02;5.00386420e-02;4.35746710e-02;3.67106700e-02;7.76113317e-03
-8.50120830e-03;-9.45628510e-03;1.62262700e-03;9.89043910e-03;1.37687800e-02;1.36009500e-02;9.17248980e-03;-5.09486650e-03;-3.96293600e-02;-
9.41107350e-02;-1.30937500e-01;-1.44678780e-01;-1.52125810e-01;-
1.62308200e-01;-1.71625890e-01;-1.78240810e-01;-1.75740190e-01;-
1.60872980e-01;-1.35654930e-01;-1.04207340e-01;-7.59180180e-02;-
5.08249220e-02;-2.70282870e-02;-5.42865100e-03;1.43919560e-02;3.27009310e-02;4.87396140e-02;6.23603090e-02;7.30669390e-02;8.02614900e-02;8.44652320e-02;8.57574220e-02;8.42961350e-02;7.99435270e-02;7.40838020e-02;6.72745380e-02;6.03645520e-02;5.25514450e-02;4.55704910e-02;3.81570170e-02;7.19308378e-03
-8.42080950e-03;-9.36479230e-03;1.95402530e-03;1.04479790e-02;1.44822270e-02;1.44305220e-02;9.99445890e-03;-4.39585710e-03;-3.92232580e-02;-
9.41034660e-02;-1.31273500e-01;-1.45475590e-01;-1.53385820e-01;-
1.64082040e-01;-1.73828990e-01;-1.80745110e-01;-1.78283370e-01;-
1.63356540e-01;-1.37847120e-01;-1.06002860e-01;-7.73606110e-02;-
5.19688510e-02;-2.78865540e-02;-6.05507300e-03;1.40050320e-02;3.25894190e-02;4.89010270e-02;6.28192770e-02;7.38411730e-02;8.13108360e-02;8.57574220e-02;8.72420610e-02;8.58840480e-02;8.15250150e-02;7.55946300e-02;6.86618080e-02;6.15860770e-02;5.35683220e-02;4.63578100e-02;3.86914310e-02;6.70979102e-03
-7.65892220e-03;-8.42399950e-03;2.86783770e-03;1.12523070e-02;1.52040130e-02;1.51562160e-02;1.08018850e-02;-3.27759400e-03;-3.73222460e-02;-
9.10059510e-02;-1.27233990e-01;-1.41183150e-01;-1.49179480e-01;-
1.60073740e-01;-1.70049860e-01;-1.77146080e-01;-1.74837480e-01;-
1.60224210e-01;-1.35174910e-01;-1.03889590e-01;-7.57511280e-02;-
5.08453570e-02;-2.72455700e-02;-5.88652190e-03;1.37251560e-02;3.19234070e-02;4.78945850e-02;6.15484760e-02;7.24194510e-02;7.98271060e-02;8.42961350e-02;8.58840480e-02;8.47002690e-02;8.05258460e-02;7.47709380e-02;6.79966250e-02;6.10373820e-02;5.31293920e-02;4.59736110e-02;3.83454520e-02;6.25738742e-03
-6.40761030e-03;-6.90247320e-03;4.04730860e-03;1.19576800e-02;1.56237740e-02;1.55100830e-02;1.13775450e-02;-1.93553110e-03;-3.41123160e-02;-
8.49125980e-02;-1.18963580e-01;-1.31989310e-01;-1.39691680e-01;-
1.50351040e-01;-1.60122780e-01;-1.66990560e-01;-1.64802460e-01;-
1.50897170e-01;-1.27131960e-01;-9.74773940e-02;-7.08074380e-02;-
4.72512370e-02;-2.49716340e-02;-4.86196780e-03;1.35557610e-02;3.06573060e-02;4.56521770e-02;5.84666280e-02;6.86961320e-02;7.56843350e-02;7.99435270e-02;8.15250150e-02;8.05258460e-02;7.67178190e-02;7.13881390e-02;6.50653870e-02;5.85296280e-02;5.10681430e-02;4.42795260e-02;3.70140980e-02;6.02865638e-03
-5.12864270e-03;-5.31118480e-03;5.15381810e-03;1.24151060e-02;1.57230480e-02;1.55000460e-02;1.16793140e-02;-6.05837780e-04;-3.02759950e-02;-
7.72204280e-02;-1.08389630e-01;-1.20187480e-01;-1.27408800e-01;-
1.37622700e-01;-1.47020220e-01;-1.53543870e-01;-1.51533140e-01;-
1.38548470e-01;-1.16492170e-01;-8.90074540e-02;-6.42941900e-02;-
4.25170730e-02;-2.19720650e-02;-3.48498680e-03;1.33858820e-02;2.90441130e-02;4.27465500e-02;5.44343350e-02;6.37722790e-02;7.01584050e-02;7.40838020e-02;7.55946300e-02;7.47709380e-02;7.13881390e-02;6.66062390e-02;6.08910040e-02;5.49383010e-02;4.81066880e-02;4.18641180e-02;3.51532600e-02;5.86766152e-03
-3.89231390e-03;-3.78424280e-03;6.05647540e-03;1.25416940e-02;1.54052920e-02;1.50409740e-02;1.15500350e-02;4.27512070e-04;-2.63884050e-02;-
6.89423180e-02;-9.68248460e-02;-1.07195110e-01;-1.13799120e-01;-
1.23400130e-01;-1.32260580e-01;-1.38338230e-01;-1.36513950e-01;-
1.24582920e-01;-1.04473280e-01;-7.94528890e-02;-5.69613850e-02;-

3.71925950e-02;-1.86006040e-02;-1.93557420e-03;1.31914090e-02;2.72093450e-02;3.94338630e-02;4.98200920e-02;5.81092630e-02;6.37724120e-02;6.72745380e-02;6.86618080e-02;6.79966250e-02;6.50653870e-02;6.08910040e-02;5.58870260e-02;5.06350110e-02;4.45674820e-02;3.90101810e-02;3.30019300e-02;5.83916581e-03
-2.86429520e-03;-2.48403250e-03;6.71376760e-03;1.24147980e-02;1.48232520e-02;1.42759200e-02;1.10781540e-02;1.05355580e-03;-2.30340310e-02;-6.13698170e-02;-8.61107340e-02;-9.50340250e-02;-1.00968630e-01;-1.09861610e-01;-1.18070230e-01;-1.23551680e-01;-1.21804390e-01;-1.10907300e-01;-9.27142490e-02;-7.01144850e-02;-4.98148530e-02;-3.20232140e-02;-1.53409130e-02;-4.56913550e-04;1.29664570e-02;2.53703130e-02;3.61315090e-02;4.52101860e-02;5.24216370e-02;5.73261140e-02;6.03645520e-02;6.15860770e-02;6.10373820e-02;5.85296280e-02;5.49383010e-02;5.06350110e-02;4.61203640e-02;4.08654610e-02;3.60622320e-02;3.08312720e-02;6.09296394e-03
-2.06116170e-03;-1.41684620e-03;7.02889770e-03;1.18762980e-02;1.37921340e-02;1.30606990e-02;1.01975480e-02;1.43039060e-03;-1.95120860e-02;-5.29457550e-02;-7.41277180e-02;-8.15191680e-02;-8.66769670e-02;-9.46554480e-02;-1.02051060e-01;-1.06877320e-01;-1.05297520e-01;-9.56169360e-02;-7.96258660e-02;-5.97982540e-02;-4.20097760e-02;-2.64632850e-02;-1.19200510e-02;9.92673720e-04;1.25525930e-02;2.31793400e-02;3.23259360e-02;3.99595680e-02;4.59768080e-02;5.00386420e-02;5.25514450e-02;5.35683220e-02;5.31293920e-02;5.10681430e-02;4.81066880e-02;4.45674820e-02;4.08654610e-02;3.65401390e-02;3.26104890e-02;2.82883620e-02;6.37213622e-03
-1.55041830e-03;-6.89173660e-04;7.05130740e-03;1.10803550e-02;1.24874820e-02;1.15020730e-02;8.87087600e-03;1.11864690e-03;-1.72809990e-02;-4.67017940e-02;-6.48370350e-02;-7.06370380e-02;-7.48487060e-02;-8.17857940e-02;-8.82558410e-02;-9.24021210e-02;-9.09060160e-02;-8.23066640e-02;-6.82574230e-02;-5.08571880e-02;-3.52636350e-02;-2.16525860e-02;-8.93427200e-03;2.30860050e-03;1.22743280e-02;2.13508020e-02;2.90665430e-02;3.53963980e-02;4.03082210e-02;4.35746710e-02;4.55704910e-02;4.63578100e-02;4.59736110e-02;4.42795260e-02;4.18641180e-02;3.90101810e-02;3.60622320e-02;3.26104890e-02;2.95601890e-02;2.61727970e-02;7.08700565e-03
-1.10416420e-03;-3.38528910e-05;6.91493550e-03;1.00762980e-02;1.09206240e-02;9.66638330e-03;7.25725550e-03;5.60643830e-04;-1.52587280e-02;-4.05531580e-02;-5.55341740e-02;-5.95539630e-02;-6.25845430e-02;-6.82387580e-02;-7.36124940e-02;-7.70454290e-02;-7.56910460e-02;-6.83109820e-02;-5.63680070e-02;-4.15764860e-02;-2.83239190e-02;-1.67489360e-02;-5.93385850e-03;3.58258320e-03;1.18969330e-02;1.93503580e-02;2.55672120e-02;3.05331150e-02;3.42810450e-02;3.67106700e-02;3.81570170e-02;3.86914310e-02;3.83454520e-02;3.70140980e-02;3.51532600e-02;3.30019300e-02;3.08312720e-02;2.82883620e-02;2.61727970e-02;2.38052950e-02;7.94361408e-03
3.71254550e-03;3.41629837e-03;5.17981497e-03;5.01996838e-03;4.01331378e-03;2.09588537e-03;4.68012767e-04;-2.76679346e-03;-1.07124776e-02;-2.29238276e-02;-2.70210087e-02;-2.21019300e-02;-1.67392046e-02;-1.30619144e-02;-1.07139874e-02;-9.12976283e-03;-7.89632209e-03;-6.76772549e-03;-5.14296457e-03;-2.85893296e-03;-4.45332526e-04;1.82265152e-03;4.09821215e-03;6.00248266e-03;7.41624628e-03;8.31022897e-03;8.79772472e-03;8.82674905e-03;8.36643346e-03;7.76113317e-03;7.19308378e-03;6.70979102e-03;6.25738742e-03;6.02865638e-03;5.86766152e-03;5.83916581e-03;6.09296394e-03;6.37213622e-03;7.08700565e-03;7.94361408e-03;1.10840927e-02