Universidad Zaragoza
1542

Proyecto Fin de Carrera

Optimización del mejoramiento de crudos pesados mediante simulación fluidodinámica en condiciones de cavitación ultrasónica.

Autor

Fernando Gracia Guinaldo

Director

Javier Blasco Alberto

Escuela de Ingeniería y Arquitectura

2012

ANEXO I

1. O.V. Abramov, V.O. Abramov, S.K. Myasnikov, M.S.Mullakaev. Extraction of bitumen, crude oil and its products from tar sand and contaminated sandy soil under effect of ultrasound. October 2008.
2. Lekhraj P. Amina, Parag R. Gogateb, Arthur E. Burgessa, David H.Bremnera. Optimization of a hydrodynamic cavitation reactor using salicylic acid dosimetry. September 2009.
3. Área de mecánica de fluidos Universidad de Zaragoza. "OBTENCIÓN DE DATOS TÉCNICOS BÁSICOS PARA EL MEJORAMIENTO DE CRUDOS PESADOS Y RESIDUOS DE REFINERÍA, BASADOS EN CAVITACIÓN ULTRASÓNICA E HIDRODINÁMICA.FASE I". Informe para REPSOL YPF. Diciembre 2006.
4. Área de mecánica de fluidos Universidad de Zaragoza. "EVALUACIÓN DE TECNOLOGÍAS BASADAS EN CAVITACIÓN HIDRODINÁMICA Y ESTIMULACIÓN MEDIANTE ULTRASONICDOS PARA EL MEJORAMIENTO DE CRUDOS PESADOS Y RESUDUOS DE REFINERÍA". Informe para REPSOL YPF. Septiembre 2008.
5. Arrojo Magallón Sergio, Cavitación hidrodinámica aplicada a tratamiento de efluentes contaminados., tesis doctoral, Mayo 2007.
6. César Dopazo. ¿Cavitar o no cavitar? La inevitable ubicuidad de las burbujas. Discurso inaugural del curso académico 2008 de la Real Academia de Ingeniería.
7. Kai Dunn, TehFu Yen. A plausible reaction pathway of asphaltene under ultrasound. February 2001.
8. Daniel Fuster. Simulación de la cavitación hidrodinámica. Proyecto fin de carrera. Junio 2003.
9. Daniel Fuster. Modelling and Numerical Simulation of the Dynamics of Liquid-Bubble Cavitating Systems with Chemical Reaction Processes. Abril 2007.
10. Parag R.Gogate. Cavitation: an auxiliary technique in wastewater treatment schemes. June 2002.
11. Parag R. Gogate. Cavitational reactors for process intensification of chemical processing applications: A critical review. September 2007.
12. J.O. Hirschfelder, C.F. Curtiss, R.B. Bird, Molecular, Wiley. Theory of Gases and Liquids, New York, 1954.
13. Parag M. Kanthale, Parag R. Gogate, Aniruddha B. Pandit, Anne Marie Wilhelm. Dynamics of cavitational bubbles and design of a hydrodynamic cavitational reactor: cluster approach. May 2004.
14. S. Kenneth Suslick, Nathan C. Eddingsaas, David J. Flannigan, Stephen D. Hopkins, Hangxun Xu. Extreme conditions during multibubble cavitation: Sonoluminescence as a spectroscopic probe. Department of Chemistry, University of Illinois. December 2010.
15. Ramesh Kuppa, Vijayanand S. Moholkar. Physical features of ultrasound-enhanced heterogeneous permanganate oxidation. Department of Chemical Engineering, Indian Institute of Technology Guwahati. May 2009.

16. Maria T. Martinez, Ana M. Benito and Maria A. Callejas. Thermal cracking of coal residues: kinetics of asphaltene decomposition. January 2007.
17. Bonnie J.McBride, Sanford Gordon, Martin A.Reno. Coefficients for calculating thermodynamic and transport properties of individual species 1993.NASA
18. V.S. Moholkar, P. Senthil Kumar, A.B. Pandit. Hydrodynamic cavitation for sonochemical effects. January 1999.
19. V. S. Moholkar, A. B. Pandit. Numerical investigations in the behaviour of one-dimensional bubbly flow in hydrodynamic cavitation. January 2001.
20. V. S. Moholkar, A. B. Pandit. Modeling of hydrodynamic cavitation reactors. June 2001.
21. Pulikesi Murugan, Nader Mahinpey, Thilakavathi Mani. Thermal cracking and combustion kinetics of asphaltenes derived from Fosterton oil. June 2009.
22. B.A. Souza, E.M. Matos, R. Guirardello, J.R. Nunhez. Predicting coke formation due to thermal cracking inside tubes of petrochemical fired heaters using a fast CFD formulation.. November 2005
23. Vinayak S. Sutkara, Parag R. Gogatea, Levente Csokab. Theoretical prediction of cavitational activity distribution in sonochemical reactors. January 2010.
24. R.C. Reid, J.M. Prausnitz, B.E. Poling. Properties of Gases and Liquids, McGrawHill, New York, 1987.
25. Jinsheng Wang, Edward J. Anthony. A study of thermal-cracking behavior of asphaltenes. Chemical Engineering Science, Volume 58, Issue 1, Pages 157-162. January 2003.
26. A. Werner, J. C. de Hemptinne, F. Behar, E. Behar, C. Boned. A new viscosity model for petroleum fluids with high asphaltenes content. Fluid Phase Equilibria, Volume 147, Issues 1-2, Pages 319-341, June 1998.
27. Yingxian Zhao, Ying Yu. Kinetics of asphaltene thermal cracking and catalytic hydrocracking. Fuel Processing Techonology, Volume 92, Issue 5, Pages 977-982, January 2011.

Anexo II

A continuación se muestra el programa informático desarrollado en este Proyecto. El lenguaje de programación utilizado es C++.

```
# include <iostream>
# include <cmath>
# include <algorithm>
#include <fstream>
template <class ForwardIterator>
  ForwardIterator max_element ( ForwardIterator first, ForwardIterator last )
{
  ForwardIterator largest = first;
  if (first==last) return last;
  while (++first!=last)
    if (*largest<*first)
      largest=first;
  return largest;
}
template <class ForwardIterator>
  ForwardIterator min_element ( ForwardIterator first, ForwardIterator last )
{
  ForwardIterator lowest = first;
  if (first==last) return last;
  while (++first!=last)
    if (*first<*lowest)
      lowest=first;
  return lowest;
}
long double f1 (long double s_old)
   {
            return s_old ;
    }
long double f2 (long double r_old,long double s_old,long double NW_old,long double NN,long double NO,long double nR,long double nDWN, long double nDWO)
     {
// This function calculates the derivative dNW/dt, i.e. the rate of change of water molecules inside the bubble.
   long double V,CN,CO,CW,xiN,xiO,xiW,DWN,DWO,D,NW,ld1,ld2,ld;
   long double pi = 3.14159265359;
            V = (4./3.)*pi*(pow(r_old,3));
            CN = NN/V;
            CO = NO/V;
            NW = NW_old;
            CW = NW/V;
            xiN = CN/(CN+nR+CO);
            xiO = CO/(CN+CO+nR);
            xiW = nR/(CN+CO+nR);
            DWN = nDWN/(nR + CN);
            DWO = nDWO/(nR + CO);
            D = 1./((xiN/((1.- xiW)*DWN)) + (xiO/((1. - xiW)*DWO)));
            ld1 = sqrt(r_old*D/fabs(s_old));
            ld2 = r_old/pi;
            long double myints[]={ld1,ld2};
```

```
                ld = *min_element (myints,myints+2);
                return 4.*pi*(pow(r_old,2))*D*(nR - CW)/ld;
            }
        long double f3 (long double r_old,long double s_old,long double NW_old,long double
t_old,long double b_1,long double NN,long double NO,long double nR,long double lambW,long
double lambN,long double lambO, long double CpW,long double CpN, long double CpO,long double
phiwo,long double phiww,long double phiwn,long double phinn,long double phino,long double
phinw,long double phioo,long double phiow,long double phion,long double hW )
            }
        long double
        xiW,xiN,xiO,dQdt,dNWdt,V,temp,Cpmix,P,dVdt,CO,CW,CN,CVW,CVO,CVN,CV,sum,chi,
th_1,lth_2,lth,Ntot,NC_tot,lambda,eW,tN,tW1,tW2,tW3,tO,NW;
        long double pi = 3.14159265359,
                B = 5.0e-29,
                k = 1.38066e-23,
                t0 = 298.15,
                TH2O1 = 2295.,
                TH2O2 = 5255.,
                TH2O3 = 5400.,
                TO = 2273.,
                TN = 3350.;
                temp = t_old;
                dNWdt = b_1;
                NW = NW_old;
                Ntot = NN + NO + NW_old;
                V = (4./3.)*pi*(pow(r_old,3.));
                CO = NO/V;
                CN = NN/V;
                CW = NW/V;
                Cpmix = CpW*nR + CpN*CN + CpO*CO;
                xiW = nR/(nR + CN + CO);
                xiN = CN/(nR + CN + CO);
                xiO = CO/(nR + CN + CO);
                NC_tot = Ntot/V;
                P = NC_tot*k*temp/(1. - NC_tot*B);
                dVdt = 4.*pi*(pow(r_old,2.))*s_old;
        // Calculating specific energies of the various species.
                eW=3.*k*temp+k*TH2O1/(exp(TH2O1/temp)1.)+k*TH2O2/(exp(TH2O2/temp)-1.)+
k*TH2O3/(exp(TH2O3/temp)-1.);
        // Calculation of heat capacities, starting with calculation of vibrational contribution to the
overall heat capacity.
                tW1= (exp(TH2O1/temp)*(pow((TH2O1/temp),2)))/(pow((exp(TH2O1/temp)-1.),2));
                tW2= (exp(TH2O2/temp)*(pow((TH2O2/temp),2)))/(pow((exp(TH2O2/temp)-1.),2));
                tW3= (exp(TH2O3/temp)*(pow((TH2O3/temp),2)))/(pow((exp(TH2O3/temp)-1.),2));
                tN = (exp(TN/temp)*(pow((TN/temp),2)))/(pow((exp(TN/temp)-1.),2));
                tO = (exp(TO/temp)*(pow((TO/temp),2)))/(pow((exp(TO/temp)-1.),2));
                CVW = NW*k*(3. + tW1 + tW2 + tW3);
                CVN = NN*k*(5./2. + tN);
                CVO = NO*k*(5./2. + tO);
                CV = CVW + CVN + CVO;
        //Calculation of the conductivity of Air - Water vapor mixture

          sum=xiW*(phiww+phiwn+phiwo)+xiN*(phinn+phino+phinw)+xiO*(phioo+phiow+phion);
          lambda = (xiW*lambW + xiN*lambN + xiO*lambO)/sum;
```

```
//Calculation of heat loss by conduction out of the bubble.
        chi = lambda/Cpmix;
        lth_1 = sqrt(r_old*chi/fabs(s_old));
        lth_2 = r_old/pi;
        long double myints[]={lth_1,lth_2};
        lth = *min_element (myints,myints+2);
        dQdt = 4.*pi*(pow(r_old,2.))*lambda*(t0 - temp)/lth;
// summing up everything to get a expression of variation of temperature inside the bubble.
return (dQdt - P*dVdt + (hW - eW)*dNWdt)/CV;
    }
long double f4 (long double t,long double r_old,long double s_old,long double NW_old,long double t_old,long double b_1,long double c_1,long double NN,long double NO)
    {
This function calculates the derivative d2R/dt2. Note that input parameters are not only the time, radius, bubble wall velocity (dR/dt), number of water molecules and temperature; but also the derivatives dNW/dt and dT/dt. These derivatives are calculated earlier at each step.
  long double  y,pt,V,pg,dNTdt,dNWdt,dTdt,dpgdt,Ntot,NC_tot,temp;
  long double       ro = 1000.,
          st = 0.0634,
          c = 1481.,
          nu = 1e-6,
          pi = 3.14159265359,
          pa = 1.*101325.,
          p0= 1.*101325.,
          freq = 150*1000.,
                k = 1.38066e-23,
          B = 5.0e-29;

        temp = t_old;
        dNWdt = b_1;
        dTdt = c_1;
        y = 2.*pi*freq*t;
        pt = p0 - pa*sin(y);
        if (pt<0)
        {
           pt=0;
        }
        Ntot = NN + NO + NW_old;
        V = (4./3.)*pi*(pow(r_old,3.));
        NC_tot = Ntot/V;
        pg = NC_tot*k*temp/(1. - NC_tot*B);
        dNTdt = dNWdt/V - (3.*NC_tot*s_old)/r_old;
        dpgdt=(dNTdt*k*temp+NC_tot*k*dTdt)/(1.-NC_tot*B)
+(NC_tot*k*temp*B*dNTdt)/(pow((1.- NC_tot*B),2.));
        return ((1. + s_old/c)/(r_old*ro*(1.-s_old/c)))*(pg - pt) +(1./(ro*c*(1.-
s_old/c)))*dpgdt - 4.*nu*s_old/((1.- s_old/c)*(pow(r_old,2.))) -2.*st/(ro*(1.-
s_old/c)*(pow(r_old,2.))) - 3.*(pow(s_old,2.))*(1.-s_old/(3.*c))/(2.*r_old*(1.-s_old/c));
    }
long double cinetica1(long double t_old,long double Csat0)
{
   long double T,k1,R,vdest;
// Universal constant of ideal gases
        R = 8.314;
// Temperature
        T = t_old;
// Kinetic parameters of thermal cracking
        k1 = (1./3600.) *(exp(29.9-(175000/(R*T))));
```

```
// Equation of thermal cracking
      vdest = k1* Csat0;
    return vdest;
}
long double cinetica2(long double t_old, long double Csat0,long double Carom0)
{
   long double T,k1,k2,R,vsat;
// Universal constant of ideal gases
          R = 8.314;
// Temperature
          T = t_old;
// Kinetic parameters of thermal cracking
          k1 = (1./3600.) *(exp(29.9-(175000/(R*T))));
          k2 = (1./3600.) *(exp(23.43-(136000/(R*T))));
// Equation of thermal cracking
      vsat = k2* Carom0 - k1*Csat0;
    return vsat;
}
long double cinetica3(long double t_old, long double Carom0,long double Cresins0)
{
   long double T,k2,k3,R,varom;
// Universal constant of ideal gases
          R = 8.314;
// Temperature
          T = t_old;
// Kinetic parameters of thermal cracking
          k2 = (1./3600.) *(exp(23.43-(136000/(R*T))));
          k3 = (1./3600.) *(exp(13.57-(85000/(R*T))));
// Equation of thermal cracking
      varom = k3* Cresins0 - k2*Carom0;
    return varom;
}
long double cinetica4(long double t_old,long double Cresins0,long double Csolasf0,long
double Cmesoasf0)
{
   long double T,k3,k4,k5,R,vresins;
// Universal constant of ideal gases
          R = 8.314;
// Temperature
          T = t_old;
// Kinetic parameters of thermal cracking
          k3 = (1./3600.) *(exp(13.57-(85000/(R*T))));
          k4 = (1./3600.) *(exp(17.07-(96000/(R*T))));
          k5 = (1./3600.) *(exp(19.34-(103000/(R*T))));
// Equation of thermal cracking
      vresins = k5* (Csolasf0 + Cmesoasf0) - (k3+2*k4) * Cresins0;
    return vresins;
}
long double cinetica5(long double t_old,long double Csolasf0,long double Cresins0)
{
   long double T,k4,k5,R,vsolasf;
// Universal constant of ideal gases
          R = 8.314;
// Temperature
```

```
            T = t_old;
// Kinetic parameters of thermal cracking
            k4 = (1./3600.) *(exp(17.07-(96000/(R*T))));
            k5 = (1./3600.) *(exp(19.34-(103000/(R*T))));
// Equation of thermal cracking
      vsolasf = k4* Cresins0 - k5* Csolasf0;
    return vsolasf;
}
long double cinetica6(long double t_old,long double Cmesoasf0,long double Cresins0)
{
   long double T,k4,k5,k6,R,vmesoasf;
// Universal constant of ideal gases
            R = 8.314;
// Temperature
            T = t_old;
// Kinetic parameters of thermal cracking
            k4 = (1./3600.) *(exp(17.07-(96000/(R*T))));
            k5 = (1./3600.) *(exp(19.34-(103000/(R*T))));
            k6 = (1./3600.) *(exp(27.54-(168000/(R*T))));

// Equation of thermal cracking
      vmesoasf = k4* Cresins0 -(k5+k6)* Cmesoasf0;

    return vmesoasf;
}
long double cinetica7(long double t_old, long double Cmesoasf0)
{
   long double T,k6,R,vcoque;
// Universal constant of ideal gases
            R = 8.314;
// Temperature
            T = t_old;
// Kinetic parameters of thermal cracking
            k6 = (1./3600.) *(exp(27.54-(168000/(R*T))));
// Equation of thermal cracking
      vcoque = k6*Cmesoasf0;
    return vcoque;
}
int main ()

{
// Physical parameters.
  long double         st = 0.0634,
               c = 1481.,
                        t0 = 40. + 273.15,
                         pi = 3.14159265359,
              p0= 1.*101325.,
            freq = 150*1000.,
            r0 = 5.0e-6,
            k = 1.38066e-23,
            a = 1.4378e-3,
            mW = 0.018/6.022e23,
            mO = 0.032/6.022e23,
            mN = 0.028/6.022e23,
            diamW = 2.65e-10,
            diamN = 3.68e-10,
            diamO = 3.43e-10,
```

```
                B = 5.0e-29,
                ekW = 380.,
                ekN = 92.,
                ekO = 113.,
                dfW = 6.,
                dfN = 5.,
                dfO = 5.;

        //Concentrations of pseudo-components
          long double   Cdest0 = 0.001,
                        Csat0 = 0.166,
                        Carom0= 0.166,
                        Cresins0 = 0.166,
                        Csolasf0 = 0.25,
                        Cmesoasf0 = 0.25,
                        Ccoque0 = 0.001;
        // Parameters of the numerical method.

          long double   a2 = 0.2, a3 = 0.3, a4 = 0.6, a5 = 1., a6 = 0.875,
                        b21 =  0.2, b31 = 3./40., b32 = 9./40.,
                        b41 =0.3, b42 = -0.9, b43 = 1.2,
                        b51=-11./54.,b52=2.5,b53=-70./27.,b54=35./27.,

b61=1631./55296.,b62=175./512.,b63=575./13824.,b64=44275./110592.,b65=253./4096.,c1=37./378.,c3=250./621.,c4=125./594.,c5=0.,c6=512./1771.,dc1=2825./27648.,dc2 = 0.,c2 = 0.,dc3=18575./48384.,dc4=13525./55296.,dc5=277./14336.,dc6=0.25 TINY=1.e-40, eps = 1.e-7, safety=0.9, pgrow=-0.2, pshrnk=-0.25, errcon=1.89e-4;
        // Variables.
          Longdouble
pvp,nR,NAir,NN,NO,diamWN,diamWO,ekWN,ekWO,muWN,muWO,thdiff,argWN,argWO,argW,argN,argO,hW,nDWN,nDWO,etaW,lambW,etaO,lambO,phioo,x1ww,x2ww,phiww,x1wn,x2wn,phiwn,x1wo,x2wo,phiwo,x1nw,x2nw,phinw,x1no,x2no,phino,x1nn,x2nn,phinn,x1ow,phiow,x1on,x2on,phion,CpW,CpN,CpO,etaN,lambN,x2ow,x1oo,x2oo,auxt,o11WN,o11WO,o22W,o22O,o22N,a_1, a_2, a_3, a_4, a_5, a_6,b_1, b_2, b_3, b_4, b_5, b_6,c_1,c_2, c_3, c_4, c_5, c_6,d_1, d_2, d_3, d_4, d_5, d_6, e_1, e_2, e_3, e_4, e_5, e_6, f_1, f_2, f_3, f_4, f_5, f_6,g_1, g_2, g_3, g_4, g_5, g_6, h_1, h_2, h_3, h_4, h_5, h_6, i_1, i_2, i_3, i_4, i_5, i_6, j_1, j_2, j_3, j_4, j_5, j_6,k_1, k_2, k_3, k_4, k_5, k_6,r_new,s_new,NW_new,t_new,Cdest_new,Csat_new,Carom_new,Cresins_new,Csolasf_new,Cmesoasf_new,Ccoque_new,Cresins4_new,Carom4_new,Csolasf4_new,Cmesoasf4_new,Ccoque4_new,Cdest4_new,Csat4_new,r4_new,s4_new,NW4_new,t4_new,Cdestt,Csatt,Cresinst,Caromt,Csolasft,Cmesoasft,Ccoquet,Tt,NWt,acc,Cdest_scal,Csat_scal,Carom_scal,Csolasf_scal,Cresins_scal,Cmesoasf_scal,Ccoque_scal,r_scal,s_scal,NW_scal,t_scal,hnext,htemp,r_err,s_err,NW_err,t_err,Cdest_err,Csat_err,Carom_err,Cresins_err,Csolasf_err,Cmesoasf_err,Ccoque_err,ratio, machno,time;
        // Output variables.
          long double  VR,NT,PG,PG_graf;
          std::ofstream f;
          f.open("fichero.txt", std::ofstream::out);
          f <<"time       ratio       NW_old          t_old           PG          s_old     ";
          f <<"\n-----------------------------------------------------------------------------------------------------------------------------------------";

          std::ofstream f22;
          f22.open("cinetica.txt", std::ofstream::out);
          f22 <<"time      Cdest       Csat        Carom          Cresins          Csolasf          Cmesoasf
Ccoque     ";
```

```
        f22 <<"\n----------------------------------------------------------------------------------------------------------------------------
----------";
        f22 <<"\n"<<0<<"     "<<Cdest0<<"         "<<Csat0<<"          "<<Carom0<<"
"<<Cresins0<<"          "<<Csolasf0<<"         "<<Cmesoasf0<<"        "<<Ccoque0;
        std::ofstream f33;
        f33.open("datosFFTratio.txt", std::ofstream::out);
        std::ofstream f44;
        f44.open("datosFFTtemp.txt", std::ofstream::out);

    // Physical parameters for bubble dynamics.

            pvp = (101325.*exp(18.3036-3816.44/(t0-46.13))/760.);

    // Number of particles of various species
            nR = pvp/(k*t0);
            NAir = (4.*pi/3.)*((p0 + 2.*st/r0)*(1. - a)*(pow(r0,3.)))/(k*t0);
            NN = 0.79*NAir;
            NO = 0.21*NAir;
            muWN = 2.*mW*mN/(mW + mN);
            muWO = 2.*mW*mO/(mW + mO);
            diamWN = (diamW + diamN)/2.;
            diamWO = (diamW + diamO)/2.;
            ekWN = sqrt(ekN*ekW);
            ekWO = sqrt(ekW*ekO);

    // Evaluating the collision integrals using dimensionless temperatures.
    // These integrals will be used for calculations of coefficients of viscosity, conductivity and
diffusivity.
            argWN = t0/ekWN;
            argWO = t0/ekWO;
            argW = t0/ekW;
            argN = t0/ekN;
            argO = t0/ekO;
            o11WN = 0.87301 + 2.46961*exp(-argWN/0.27359) + 1.26848*exp(-
argWN/1.07918);
            o11WO = 0.87301 + 2.46961*exp(-argWO/0.27359) + 1.26848*exp(-
argWO/1.07918);
            o22W = 0.8976 + 2.4112*exp(-argW/0.39146) + 0.95159*exp(-argW/1.55987);
            o22O= 0.8976 + 2.4112*exp(-argO/0.39146) + 0.95159*exp(-argO/1.55987);
            o22N = 0.8976 + 2.4112*exp(-argN/0.39146) + 0.95159*exp(-argN/1.55987);
    //  Calculating coefficients of diffusivity, viscosity and conductivity.
    //  The formulae are from Molecular Theory of Gases and Liquids by Hirschfelder, Curtiss and
Bird.
            hW = (1.+ dfW)*k*t0/2.;
            nDWN= (3./8.)*(pow((pi*k*t0/muWN),0.5))/(pi*(pow(diamWN,2.))*o11WN);
            nDWO= (3./8.)*(pow((pi*k*t0/muWO),0.5))/(pi*(pow(diamWO,2.))*o11WO);
            CpW = (dfW + 2.)*k/2.;
            CpN = (dfN + 2.)*k/2.,
            CpO = (dfO + 2.)*k/2.;
            etaW = (5./16.)*(pow((pi*mW*k*t0),0.5))/(pi*(pow(diamW,2.))*o22W);
            lambW = (15.*k*etaW/(mW*4.))*(4.*dfW/30. + 3./5.),
            etaN = (5./16.)*(pow((pi*mN*k*t0),0.5))/(pi*(pow(diamN,2.))*o22N);
            lambN = (15.*k*etaN/(mN*4.))*(4.*dfN/30. + 3./5.),
            etaO = (5./16.)*(pow((pi*mO*k*t0),0.5))/(pi*(pow(diamO,2.))*o22O);
            lambO = (15.*k*etaO/(mO*4.))*(4.*dfO/30. + 3./5.);

            x1ww = (pow((etaW/etaW),0.5))*(pow((mW/mW),0.25));
```

```
x2ww = pow((1. + (mW/mW)),(-0.5));
phiww = (1./2.8284)*x2ww*(pow((1. + x1ww),2.));
x1wn = (pow((etaW/etaN),0.5))*(pow((mW/mN),0.25));
x2wn = pow((1. + (mW/mN)),-0.5);
phiwn = (1./2.8284)*x2wn*(pow((1. + x1wn),2.));
x1wo = (pow((etaW/etaO),0.5))*(pow((mW/mO),0.25));
x2wo = pow((1. + (mW/mO)),-0.5);
phiwo = (1./2.8284)*x2wo*(pow((1. + x1wo),2.));

x1nw = (pow((etaN/etaW),0.5))*(pow((mN/mW),0.25));
x2nw = pow((1. + (mN/mW)),-0.5);
phinw = (1./2.8284)*x2nw*(pow((1. + x1nw),2.));
x1no = (pow((etaN/etaO),0.5))*(pow((mN/mO),0.25));
x2no = pow((1. + (mN/mO)),-0.5);
phino = (1./2.8284)*x2no*(pow((1. + x1no),2.));
x1nn = (pow((etaN/etaN),0.5))*(pow((mN/mN),0.25));
x2nn = pow((1. + (mN/mN)),-0.5);
phinn = (1./2.8284)*x2nn*(pow((1. + x1nn),2.));

x1ow = (pow((etaO/etaW),0.5))*(pow((mO/mW),0.25));
x2ow =pow((1. + (mO/mW)),-0.5);
phiow = (1./2.8284)*x2ow*(pow((1. + x1ow),2.));
x1on = (pow((etaO/etaN),0.5))*(pow((mO/mN),0.25));
x2on = pow((1. + (mO/mN)),-0.5);
phion = (1./2.8284)*x2on*(pow((1. + x1on),2.));
x1oo =(pow((etaO/etaO),0.5))*(pow((mO/mO),0.25));
x2oo = (pow((1. + (mO/mO)),-0.5));
phioo = (1./2.8284)*x2oo*(pow((1. + x1oo),2.));

long double     t = 0,
                r_old = r0,
                s_old = 1.e-8,
                 t_old = t0,
                Tref = 0,
                NW_old = 0,
                h = 1.e-8,
                hmin = 0,
                 errmax = 2,
                 auxtime = 0,
                cont=0,
                 time = t*freq;


  while (time<5000)
   {
   while (errmax>1)
   {

// Runge-Kutta-Fehlberg:
        a_1 = h*f1(s_old);
        b_1 = h*f2(r_old,s_old,NW_old,NN,NO,nR,nDWN,nDWO);
        c_1=
h*f3(r_old,s_old,NW_old,t_old,b_1,NN,NO,nR,lambW,lambN,lambO,CpW,CpN,CpO,phiwo,phiww,ph
iwn,phinn,phino,phinw,phioo,phiow,phion,hW);
```

```
d_1 = h*f4(t,r_old,s_old,NW_old,t_old,b_1,c_1,NN,NO);
e_1 = h*cinetica1(t_old,Csat0);
f_1 = h*cinetica2(t_old,Csat0,Carom0);
g_1 = h*cinetica3(t_old,Carom0,Cresins0);
h_1 = h*cinetica4(t_old,Cresins0,Csolasf0,Cmesoasf0);
i_1 = h*cinetica5(t_old,Csolasf0,Cresins0);
j_1 = h*cinetica6(t_old,Cmesoasf0,Cresins0);
k_1 = h*cinetica7(t_old,Cmesoasf0);

a_2 = h*f1(s_old + b21*d_1);
b_2=
h*f2(r_old+b21*a_1,s_old+b21*d_1,NW_old+b21*b_1,NN,NO,nR,nDWN,nDWO);
c_2=
h*f3(r_old+b21*a_1,s_old+b21*d_1,NW_old+b21*b_1,t_old+b21*c_1,b_2,NN,NO,nR,lambW,lambN
,lambO,CpW,CpN,CpO,phiwo,phiww,phiwn,phinn,phino,phinw,phioo,phiow,phion,hW);
d_2=
h*f4(t+a2*h,r_old+b21*a_1,s_old+b21*d_1,NW_old+b21*b_1,t_old+b21*c_1,b_2,c_2,NN,NO);
e_2 = h*cinetica1(t_old+b21*c_1,Csat0+b21*f_1);
f_2 = h*cinetica2(t_old+b21*c_1,Csat0+b21*f_1,Carom0+b21*g_1);
g_2 = h*cinetica3(t_old+b21*c_1,Carom0+b21*g_1,Cresins0+b21*h_1);
h_2=
h*cinetica4(t_old+b21*c_1,Cresins0+b21*h_1,Csolasf0+b21*i_1,Cmesoasf0+b21*j_1);
i_2 = h*cinetica5(t_old+b21*c_1,Csolasf0+b21*i_1,Cresins0+b21*h_1);
j_2= h*cinetica6(t_old+b21*c_1,Cmesoasf0+b21*j_1,Cresins0+b21*h_1);
k_2 = h*cinetica7(t_old+b21*c_1,Cmesoasf0+b21*j_1);

a_3 = h*f1(s_old + b31*d_1 + b32*d_2);
b_3=
h*f2(r_old+b31*a_1+b32*a_2,s_old+b31*d_1+b32*d_2,NW_old+b31*b_1+b32*b_2,NN,NO,nR,nD
WN,nDWO);
c_3=
h*f3(r_old+b31*a_1+b32*a_2,s_old+b31*d_1+b32*d_2,NW_old+b31*b_1+b32*b_2,t_old+b31*c_1
+b32*c_2,b_3,NN,NO,nR,lambW,lambN,lambO,CpW,CpN,CpO,phiwo,phiww,phiwn,phinn,phino,phin
w,phioo,phiow,phion,hW);
d_3=
h*f4(t+a3*h,r_old+b31*a_1+b32*a_2,s_old+b31*d_1+b32*d_2,NW_old+b31*b_1+b32*b_2,t_old+b
31*c_1+b32*c_2,b_3,c_3,NN,NO);
e_3 = h*cinetica1(t_old+b31*c_1+b32*c_2,Csat0+b31*f_1+b32*f_2);
f_3=
h*cinetica2(t_old+b31*c_1+b32*c_2,Csat0+b31*f_1+b32*f_2,Carom0+b31*g_1+b32*g_2);
g_3=
h*cinetica3(t_old+b31*c_1+b32*c_2,Carom0+b31*g_1+b32*g_2,Cresins0+b31*h_1+b32*h_2);
h_3=
h*cinetica4(t_old+b31*c_1+b32*c_2,Cresins0+b31*h_1+b32*h_2,Csolasf0+b31*i_1+b32*i_2,Cmeso
asf0+b31*j_1+b32*j_2);
i_3=
h*cinetica5(t_old+b31*c_1+b32*c_2,Csolasf0+b31*i_1+b32*i_2,Cresins0+b31*h_1+b32*h_2);
j_3=
h*cinetica6(t_old+b31*c_1+b32*c_2,Cmesoasf0+b31*j_1+b32*j_2,Cresins0+b31*h_1+b32*h_2);
k_3= h*cinetica7(t_old+b31*c_1+b32*c_2,Cmesoasf0+b31*j_1+b32*j_2);


a_4 = h*f1(s_old+b41*d_1+b42*d_2+b43*d_3);
b_4=
h*f2(r_old+b41*a_1+b42*a_2+b43*a_3,s_old+b41*d_1+b42*d_2+b43*d_3,NW_old+b41*b_1+b42
*b_2+b43*b_3,NN,NO,nR,nDWN,nDWO);
```

```
c_4=
h*f3(r_old+b41*a_1+b42*a_2+b43*a_3,s_old+b41*d_1+b42*d_2+b43*d_3,NW_old+b41*b_1+b42*b_2+b43*b_3,t_old+b41*c_1+b42*c_2+b43*c_3,b_4,NN,NO,nR,lambW,lambN,lambO,CpW,CpN,CpO,phiwo,phiww,phiwn,phinn,phino,phinw,phioo,phiow,phion,hW);
d_4=
h*f4(t+a4*h,r_old+b41*a_1+b42*a_2+b43*a_3,s_old+b41*d_1+b42*d_2+b43*d_3,NW_old+b41*b_1+b42*b_2+b43*b_3,t_old+b41*c_1+b42*c_2+b43*c_3,b_4,c_4,NN,NO);
e_4=
h*cinetica1(t_old+b41*c_1+b42*c_2+b43*c_3,Csat0+b41*f_1+b42*f_2+b43*f_3);
f_4=
h*cinetica2(t_old+b41*c_1+b42*c_2+b43*c_3,Csat0+b41*f_1+b42*f_2+b43*e_3,Carom0+b41*g_1+b42*g_2+b43*g_3);
g_4=
h*cinetica3(t_old+b41*c_1+b42*c_2+b43*c_3,Carom0+b41*g_1+b42*g_2+b43*g_3,Cresins0+b41*h_1+b42*h_2+b43*h_3);
h_4=
h*cinetica4(t_old+b41*c_1+b42*c_2+b43*c_3,Cresins0+b41*h_1+b42*h_2+b43*h_3,Csolasf0+b41*i_1+b42*i_2+b43*i_3,Cmesoasf0+b41*j_1+b42*j_2+b43*j_3);
i_4=
h*cinetica5(t_old+b41*c_1+b42*c_2+b43*c_3,Csolasf0+b41*i_1+b42*i_2+b43*i_3,Cresins0+b41*h_1+b42*h_2+b43*h_3);
j_4=
h*cinetica6(t_old+b41*c_1+b42*c_2+b43*c_3,Cmesoasf0+b41*j_1+b42*j_2+b43*j_3,Cresins0+b41*h_1+b42*h_2+b43*h_3);
k_4=
h*cinetica7(t_old+b41*c_1+b42*c_2+b43*c_3,Cmesoasf0+b41*j_1+b42*j_2+b43*j_3;

a_5 = h*f1(s_old+b51*d_1+b52*d_2+b53*d_3+b54*d_4);
b_5=
h*f2(r_old+b51*a_1+b52*a_2+b53*a_3+b54*a_4,s_old+b51*d_1+b52*d_2+b53*d_3+b54*d_4,NW_old+b51*b_1+b52*b_2+b53*b_3+b54*b_4,NN,NO,nR,nDWN,nDWO);
c_5=
h*f3(r_old+b51*a_1+b52*a_2+b53*a_3+b54*a_4,s_old+b51*d_1+b52*d_2+b53*d_3+b54*d_4,NW_old+b51*b_1+b52*b_2+b53*b_3+b54*b_4,t_old+b51*c_1+b52*c_2+b53*c_3+b54*c_4,b_5,NN,NO,nR,lambW,lambN,lambO,CpW,CpN,CpO,phiwo,phiww,phiwn,phinn,phino,phinw,phioo,phiow,phion,hW);
d_5=
h*f4(t+a5*h,r_old+b51*a_1+b52*a_2+b53*a_3+b54*a_4,s_old+b51*d_1+b52*d_2+b53*d_3+b54*d_4,NW_old+b51*b_1+b52*b_2+b53*b_3+b54*b_4,t_old+b51*c_1+b52*c_2+b53*c_3+b54*c_4,b_5,c_5,NN,NO);
e_5=
h*cinetica1(t_old+b51*c_1+b52*c_2+b53*c_3+b54*c_4,Csat0+b51*f_1+b52*f_2+b53*f_3+b54*f_4);
f_5=
h*cinetica2(t_old+b51*c_1+b52*c_2+b53*c_3+b54*c_4,Csat0+b51*f_1+b52*f_2+b53*f_3+b54*f_4,Carom0+b51*g_1+b52*g_2+b53*g_3+b54*g_4);
g_5=
h*cinetica3(t_old+b51*c_1+b52*c_2+b53*c_3+b54*c_4,Carom0+b51*g_1+b52*g_2+b53*g_3+b54*g_4,Cresins0+b51*h_1+b52*h_2+b53*h_3+b54*h_4);
h_5=
h*cinetica4(t_old+b51*c_1+b52*c_2+b53*c_3+b54*c_4,Cresins0+b51*h_1+b52*h_2+b53*h_3+b54*h_4,Csolasf0+b51*i_1+b52*i_2+b53*i_3+b54*i_4,Csat0+b51*f_1+b52*f_2+b53*f_3+b54*f_4);
```

```
                i_5=
h*cinetica5(t_old+b51*c_1+b52*c_2+b53*c_3+b54*c_4,Csolasf0+b51*i_1+b52*i_2+b53*i_3+b54*i
_4,Cresins0+b51*h_1+b52*h_2+b53*h_3+b54*h_4);
                j_5=
h*cinetica6(t_old+b51*c_1+b52*c_2+b53*c_3+b54*c_4,Cmesoasf0+b51*j_1+b52*j_2+b53*j_3+b54
*j_4,Cresins0+b51*h_1+b52*h_2+b53*h_3+b54*h_4);
                k_5=
h*cinetica7(t_old+b51*c_1+b52*c_2+b53*c_3+b54*c_4,Cmesoasf0+b51*j_1+b52*j_2+b53*j_3+b54
*j_4);

                a_6 = h*f1(s_old+b61*d_1+b62*d_2+b63*d_3+b64*d_4+b65*d_5);
                b_6=
h*f2(r_old+b61*a_1+b62*a_2+b63*a_3+b64*a_4+b65*a_5,s_old+b61*d_1+b62*d_2+b63*d_3+b64
*d_4+b65*d_5,NW_old+b61*b_1+b62*b_2+b63*b_3+b64*b_4+b65*b_5,NN,NO,nR,nDWN,nDWO);
                c_6=
h*f3(r_old+b61*a_1+b62*a_2+b63*a_3+b64*a_4+b65*a_5,s_old+b61*d_1+b62*d_2+b63*d_3+b64
*d_4+b65*d_5,NW_old+b61*b_1+b62*b_2+b63*b_3+b64*b_4+b65*b_5,t_old+b61*c_1+b62*c_2+
b63*c_3+b64*c_4+b65*c_5,b_6,NN,NO,nR,lambW,lambN,lambO,CpW,CpN,CpO,phiwo,phiww,phiw
n,phinn,phino,phinw,phioo,phiow,phion,hW);
                d_6=
h*f4(t+a6*h,r_old+b61*a_1+b62*a_2+b63*a_3+b64*a_4+b65*a_5,s_old+b61*d_1+b62*d_2+b63*d
_3+b64*d_4+b65*d_5,NW_old+b61*b_1+b62*b_2+b63*b_3+b64*b_4+b65*b_5,t_old+b61*c_1+b6
2*c_2+b63*c_3+b64*c_4+b65*c_5,b_6,c_6,NN,NO);
                e_6=
h*cinetica1(t_old+b61*c_1+b62*c_2+b63*c_3+b64*c_4+b65*c_5,Csat0+b61*f_1+b62*f_2+b63*f_3
+b64*f_4+b65*f_5);
                f_6=
h*cinetica2(t_old+b61*c_1+b62*c_2+b63*c_3+b64*c_4+b65*c_5,Csat0+b61*f_1+b62*f_2+b63*f_3
+b64*f_4+b65*f_5,Carom0+b61*g_1+b62*g_2+b63*g_3+b64*g_4+b65*g_5);
                g_6=
h*cinetica3(t_old+b61*c_1+b62*c_2+b63*c_3+b64*c_4+b65*c_5,Carom0+b61*g_1+b62*g_2+b63*
g_3+b64*g_4+b65*g_5,Cresins0+b61*h_1+b62*h_2+b63*h_3+b64*h_4+b65*h_5);
                h_6=
h*cinetica4(t_old+b61*c_1+b62*c_2+b63*c_3+b64*c_4+b65*c_5,Cresins0+b61*h_1+b62*h_2+b63
*h_3+b64*h_4+b65*h_5,Csolasf0+b61*i_1+b62*i_2+b63*i_3+b64*i_4+b65*i_5,Cmesoasf0+b61*j_1
+b62*j_2+b63*j_3+b64*j_4+b65*j_5);
                i_6=
h*cinetica5(t_old+b61*c_1+b62*c_2+b63*c_3+b64*c_4+b65*c_5,Csolasf0+b61*i_1+b62*i_2+b63*i
_3+b64*i_4+b65*i_5,Cresins0+b61*h_1+b62*h_2+b63*h_3+b64*h_4+b65*h_5);
                j_6=
h*cinetica6(t_old+b61*c_1+b62*c_2+b63*c_3+b64*c_4+b65*c_5,Cmesoasf0+b61*j_1+b62*j_2+b6
3*j_3+b64*j_4+b65*j_5,Cresins0+b61*h_1+b62*h_2+b63*h_3+b64*h_4+b65*h_5);
                k_6=
h*cinetica7(t_old+b61*c_1+b62*c_2+b63*c_3+b64*c_4+b65*c_5,Cmesoasf0+b61*j_1+b62*j_2+b6
3*j_3+b64*j_4+b65*j_5);

        /*New values of the variables from R-K 5th order formula*/
                r_new = r_old+ (c1*a_1+c2*a_2+c3*a_3+c4*a_4+c5*a_5+c6*a_6);
                s_new = s_old + (c1*d_1+c2*d_2+c3*d_3+c4*d_4+c5*d_5+c6*d_6);
                NW_new=NW_old+ (c1*b_1+c2*b_2+c3*b_3+c4*b_4+c5*b_5+c6*b_6);
                t_new = t_old + (c1*c_1+c2*c_2+c3*c_3+c4*c_4+c5*c_5+c6*c_6);
                Cdest_new=Cdest0+(c1*e_1+c2*e_2+c3*e_3+c4*e_4+c5*e_5+c6*e_6);
                Csat_new = Csat0 + (c1*f_1+c2*f_2+c3*f_3+c4*f_4+c5*f_5+c6*f_6);
                   Carom_new=Carom0+(c1*g_1+c2*g_2+c3*g_3+c4*g_4+c5*g_5+c6*g_6);
                Cresins_new=Cresins0+(c1*h_1+c2*h_2+c3*h_3+c4*h_4+c5*h_5+c6*h_6);
                Csolasf_new=Csolasf0 + (c1*i_1+c2*i_2+c3*i_3+c4*i_4+c5*i_5+c6*i_6);
                Cmesoasf_new=Cmesoasf0+(c1*j_1+c2*j_2+c3*j_3+c4*j_4+c5*j_5+c6*j_6);
```

```
                Ccoque_new=Ccoque0+(c1*k_1+c2*k_2+c3*k_3+c4*k_4+c5*k_5+c6*k_6);

        /*New values of the variables from R-K embedded 4th order formula*/

                r4_new=r_old+(dc1*a_1+dc2*a_2+dc3*a_3+dc4*a_4+dc5*a_5+dc6*a_6);
                s4_new=s_old+(dc1*d_1+dc2*d_2+dc3*d_3+dc4*d_4+dc5*d_5+dc6*d_6);
                NW4_new=NW_old+(dc1*b_1+dc2*b_2+dc3*b_3+dc4*b_4+dc5*b_5+dc6*b_6);
                t4_new=t_old+(dc1*c_1+dc2*c_2+dc3*c_3+dc4*c_4+dc5*c_5+dc6*c_6);
                Cdest4_new=Cdest0+(dc1*e_1+dc2*e_2+dc3*e_3+dc4*e_4+dc5*e_5+dc6*e_6);
                Csat4_new=Csat0+(dc1*f_1+dc2*f_2+dc3*f_3+dc4*f_4+dc5*f_5+dc6*f_6);
                Carom4_new=Carom0+(dc1*g_1+dc2*g_2+dc3*g_3+dc4*g_4+dc5*g_5+dc6*g_6);
                Cresins4_new=Cresins0+(dc1*h_1+dc2*h_2+dc3*h_3+dc4*h_4+dc5*h_5+dc6*h_6);
                Csolasf4_new=Csolasf0+(dc1*i_1+dc2*i_2+dc3*i_3+dc4*i_4+dc5*i_5+dc6*i_6);
                Cmesoasf4_new=Cmesoasf0+(dc1*j_1+dc2*j_2+dc3*j_3+dc4*j_4+dc5*j_5+dc6*j_6);
                Ccoque4_new=Ccoque0+(dc1*k_1+dc2*k_2+dc3*k_3+dc4*k_4+dc5*k_5+dc6*k_6);

        /*Calculating functions for scaling factors*/
                NWt = f2(r_old,s_old,NW_old,NN,NO,nR,nDWN,nDWO);
                Tt=
f3(r_old,s_old,NW_old,t_old,b_1,NN,NO,nR,lambW,lambN,lambO,CpW,CpN,CpO,phiwo,phiww,phiw
n,phinn,phino,phinw,phioo,phiow,phion,hW);
                acc = f4(t,r_old,s_old,NW_old,t_old,b_1,c_1,NN,NO);
                Cdestt = cinetica1(t_old,Csat0);
                Csatt = cinetica2(t_old,Csat0,Carom0);
                Caromt = cinetica3(t_old,Carom0,Cresins0);
                Cresinst= cinetica4(t_old,Cresins0,Csolasf0,Cmesoasf0);
                Csolasft = cinetica5(t_old,Csolasf0,Cresins0);
                Cmesoasft= cinetica6(t_old,Cmesoasf0,Cresins0);
                Ccoquet= cinetica7(t_old,Cmesoasf0);
        /*Scaling factors for error calculations*/
                r_scal = r_old + h*s_old + TINY;
                s_scal = s_old + h*acc + TINY;
                NW_scal = NW_old + h*NWt + TINY;
                t_scal = t_old + h*Tt + TINY;
                Cdest_scal = Cdest0 + h*Cdestt + TINY;
                Csat_scal = Csat0 + h*Csatt + TINY;
                Carom_scal = Carom0 + h*Caromt + TINY;
                Cresins_scal = Cresins0 + h*Cresinst + TINY;
                Csolasf_scal = Csolasf0 + h*Csolasft + TINY;
                Cmesoasf_scal = Cmesoasf0 + h*Cmesoasft + TINY;
                Ccoque_scal = Ccoque0 + h*Ccoquet + TINY;

        /*Calculation of stepsize after error analysis*/
                r_err = fabs (r_new-r4_new);
                s_err = fabs(s_new - s4_new);
                NW_err = fabs(NW_new - NW4_new);
                t_err = fabs(t_new - t4_new);
                Cdest_err = fabs(Cdest_new - Cdest4_new);
                Csat_err = fabs(Csat_new - Csat4_new);
                Carom_err = fabs(Carom_new - Carom4_new);
                Cresins_err = fabs(Cresins_new - Cresins4_new);
                Csolasf_err = fabs(Csolasf_new - Csolasf4_new);
                Cmesoasf_err = fabs(Cmesoasf_new - Cmesoasf4_new);
                Ccoque_err = fabs(Ccoque_new - Ccoque4_new);
```

```
                errmax=0.;

longdoublemyints_2[]={errmax,fabs(r_err/r_scal),fabs(s_err/s_scal),fabs(t_err/t_scal),fabs(NW_err/
NW_scal),fabs(Cdest_err/Cdest_scal),fabs(Csat_err/Csat_scal),fabs(Carom_err/Carom_scal),fabs(Cre
sins_err/Cresins_scal),fabs(Csolasf_err/Csolasf_scal),fabs(Cmesoasf_err/Cmesoasf_scal),fabs(Ccoque
_err/Ccoque_scal)};
                errmax=*max_element(myints_2,myints_2+12);
                errmax = errmax/eps;
                htemp=safety*h*(pow(errmax,pshrnk));
                long double myints[]={fabs(htemp),0.1*fabs(h)};
                h=*max_element(myints,myints+2);
                hnext=4*h;
             }
            if (errmax>errcon)
             {
                hnext=safety*h*(pow(errmax,pgrow));
             }
           if (hnext<hmin)
             {
                std::cout<<"Minium step size reacherd in the iteration";
             }
        //Resetting values for the next iteration.
                errmax=2;
                r_old = r_new;
                s_old = s_new;
                NW_old = NW_new;
                t_old = t_new;
                Cdest0=Cdest_new;
                Csat0=Csat_new;
                Carom0=Carom_new;
                Cresins0=Cresins_new;
                Csolasf0=Csolasf_new;
                Cmesoasf0=Cmesoasf_new;
                Ccoque0=Ccoque_new;
        // Calculating the pressure inside the bubble
                VR = (4./3.)*pi*pow(r_old,3.);
                NT = NN + NO + NW_old;
                PG = NT*k*t_old/(VR*(1. - NT*B));
        //Resetting the time step
                h = hnext;
                t = t + h;
        //Normalizing the variables or graph
                ratio = r_new/r0;
                machno = fabs(s_new/c);
                time = t*freq;
                Tref=Tref+((1./3600.)*(exp(29.9-(175000/(8.314*t_old))))*(time- auxtime));
                auxtime=time;
                thdiff = sqrt((1.434e-4*r_old/fabs(s_old)));
                PG_graf = PG/101325.;
                auxt=h;
                std::cout<<("\n")<<time;
                cont=cont+1;
           f <<"\n"<<time<<"      "<<ratio<<"       "<<NW_old<<"        "<<t_old<<"
"<<PG_graf<<"          "<<s_old;
```

```
            f22 <<"\n"<<time<<"     "<<Cdest_new<<"         "<<Csat_new<<"
"<<Carom_new<<"            "<<Cresins_new<<"          "<<Csolasf_new<<"        "<<Cmesoasf_new<<"
"<<Ccoque_new;
                    if (cont<65537)
            {
                    f33 <<"\n"<<time<<" "<<ratio;
            }
            if (cont<65537)
            {
                    f44 <<"\n"<<time<<" "<<t_old;
            }
        }
            f.close();
            f22.close();
            f33.close();
            f44.close();
            Tref=Tref/time;
            std::cout<<("\n")<<Tref;
            system ("PAUSE");
            return 0;
        }
```

ANEXO III

En este anexo se describen todas las variables utilizadas en las ecuaciones del modelo utilizado descrito en el posterior anexo.

Variables:

c: velocidad de sonido en el líquido [m/s]

c_p: capacidad calorífica específica molecular a presión constante [J/K]

$c_{p,mix}$: capacidad calorífica específica de una mezcla de especies a presión constante [J moléculas/K m^3]

c_v :capacidad calorífica específica molecular a volumen constante [J/K]

$c_{v,mix}$:capacidad calorífica específica de una mezcla de especies a volumen constante [J moléculas/K m^3]

c_w :concentración de moléculas de agua en la burbuja [moléculas/m^3]

c_{wR} :concentración de moléculas de agua en la interfase de la burbuja [moléculas/m^3]

C_i : concentración de especies del craqueo [kg/m^3]

D_w: coeficiente de difusión de vapor de agua [m^2/s]

D_{ij} : coeficiente de difusión de la especie i en la mezcla [(m^5/(moléculas s)]

D_{ij}^*: coeficiente de difusión de la especie i en la mezcla [m^2/s]

e/k: constantes de fuerza de Lennard-Jones [K]

f : grados de libertad de una especie [adimensional]

freq: frecuencia [Hz]

h: radio interno de van de Waals [m]

h_w: entalpia específica molecular del agua [J]

i,j: agua, nitrógeno u oxígeno

k: constante de Boltzmann [J/K]

K: constante cinética [s^{-1}]

l_{diff}: profundidad de la difusión másica [m]

l_{th}: profundidad de la difusión térmica [m]

m: masa molecular de la especie [g/molécula]

n_i: concentración de la especie i en la interfase de la burbuja [moléculas/m^3]

N_{N2}: número de moléculas de nitrógeno en la burbuja [moléculas]

N_{O2}: número de moléculas de oxigeno en la burbuja [moléculas]

N_w: número de moléculas de agua en la burbuja [moléculas]

N_{tot}: número total de moléculas en la burbuja [moléculas]

P_A: amplitud de presión [Pa]

P_i: presión dentro de la burbuja [Pa]

P_0: presión atmosférica [Pa]

P_t: presión del líquido [Pa]

Q: transferencia de calor a través de la pared de la burbuja [J/s]

R: radio de la burbuja [m]

R: constante universal de los gases [J/(mol K)]

R_0: radio inicial de la burbuja [m]

t: tiempo [s]

t_{osc}: escala temporal de oscilación de la burbuja [s]

T: temperatura en la burbuja [K]

T_0: temperatura ambiente en el líquido [K]

T_{ij}^*: temperatura reducida de dos especies [adimensional]

U_w: energía interna específica de las moléculas de agua [J]

V: volumen de la burbuja [m^3]

v: velocidad de componentes del craqueo. [kg/(m^3s)]

γ: constante politrópica del contenido de la burbuja [adimensional]

ε_i: fracción molar de la especie i en la burbuja [adimensional]

η_i: coeficiente de viscosidad de la especie i [m^2/s]

λ_i: coeficiente de conductividad térmica de la especie i [W/(K m)]

λ_{mix}: coeficiente de conductividad térmica de la mezcla de especies [W/(K m)]

ν: viscosidad cinemática de el líquido [m^2/s]

μ_{ij}: masa molecular reducida de dos especies [g/molécula]

Ω_{ij}: corrección adimensional de la desviación de la colisión de la desviación de colisión [adimensional]

ρ_L: densidad del líquido [kg/m^3]

ρ_i: densidad molecular de la especie i [moléculas/m^3]

ρ_{mix}: densidad molecular de la mezcla de especies [moléculas/m^3]

σ: tensión superficial del líquido [N/m]

σ_{ij}: parámetro de la función característica potencial de interacción ente dos especies [m]

σ_i: diámetro molecular de la especie i [m]

κ : difusividad térmica de la burbuja [m^2/s]

θ_i: temperatura vibracional característica de la especie i [K]

Anexo IV

En este anexo se mostrarán las ecuaciones que describen el modelo matemático para el movimiento radial de la burbuja de cavitación en el programa informático, así como las ecuaciones de transferencia de calor,del flujo difusivo de moléculas de agua y el balance global de calor, también se incluyen los valores iniciales tomados para su resolución. También se muestra el modelo cinético empleado. Además se detalla la fórmula del método numérico utilizado y los coeficientes del método (Runge-Kutta-Felhberg).

Movimiento de la burbuja descrito por la ecuación de Keller-Miskis:

$$\left(1-\frac{dR/dt}{c}\right)R\frac{d^2R}{dt^2}+\frac{3}{2}\left(1-\frac{dR/dt}{3c}\right)\left(\frac{dR}{dt}\right)^2=\frac{1}{\rho_L}\left(1-\frac{dR/dt}{c}\right)(\rho_L-P_t)+\frac{R}{\rho_L c}\frac{dP_i}{dt}-4\nu\frac{dR/dt}{R}-\frac{2\sigma}{\rho_L R}$$

$$P_i=\frac{N_{tot}(t)kT}{[4\pi(R^3(t)-h^3)/3]^\gamma}$$

$$\gamma=1$$

$$N_{tot}=N_{N_2}+N_{O_2}+N_w$$

$$h\sim\frac{R_0}{8.86}$$

$$P_t=P_0-P_A\sin(2\pi freqt)$$

Para t=0, el radio inicial de la burbuja, y la velocidad de la pared de la burbuja es igual a cero.

Ecuación para el flujo difusivo de vapor de agua a través de la pared de burbuja:

$$\frac{dN_w}{dt}=4\pi R^2 D_w\left(\frac{C_{wR}-C_w}{l_{diff}}\right)$$

$$l_{diff}=\min\left(\sqrt{D_w t_{osc}},\frac{R}{\pi}\right)$$

$$t_{osc} = \frac{R}{\left|\frac{dR}{dt}\right|}$$

$$D_w = \frac{1}{\frac{\varepsilon_{N_2}}{(1-\varepsilon_{N_2})D_{N_2H_2O}} + \frac{\varepsilon_{O_2}}{(1-\varepsilon_{O_2})D_{O_2H_2O}}}$$

$$D_{ij} = \frac{D_{ij}^*}{n_i + n_j}$$

$$D_{ij}^* = 3/8 \frac{\sqrt{\pi kT/\mu_{ij}}}{n_{ij}\pi\sigma_{ij}^2\Omega_{ij}^{(1,1)}}$$

$$n_{ij} = n_i + n_j$$

$$\mu_{ij} = \frac{2m_i m_j}{m_i + m_j}$$

$$\sigma_{ij} = \sigma_i + \sigma_j$$

$$\Omega_{ij}^{(n,n)} = a + b\exp\left(\frac{T_{ij}^*}{c}\right) + d\ \exp\left(\frac{T_{ij}^*}{c}\right)$$

$$T_{ij}^* = \sqrt{\frac{T^2}{(\frac{e}{k})_i(\frac{e}{k})_j}}$$

Para t=0, el número de moléculas de agua en el interior de la burbuja es cero.

<u>Ecuación de transferencia de calor a través de la burbuja:</u>

$$\frac{dQ}{dt} = 4\pi R^2\lambda(\frac{T_0 - T}{l_{th}})$$

$$l_{th} = \min(\sqrt{\kappa t_{osc}}, \frac{R}{\pi})$$

$$\kappa = \frac{\lambda}{\rho_{mix}c_{p,mix}}$$

$$\rho_{mix}c_{p,mix} = \sum_i \rho_i c_{p,i}$$

$$c_{p,i} = \frac{f+2}{2}k$$

$$\lambda = \sum_i \frac{\varepsilon_i \lambda_i}{\sum_j \varepsilon_i \phi_{i,j}}$$

$$\lambda_i = \frac{15k}{m}\eta\left(\frac{4f}{30} + \frac{3}{5}\right)$$

$$\eta = \frac{5}{16}\frac{\sqrt{\pi m k T}}{\pi \sigma_{12}^2 \Omega_{12}^{(2,2)}}$$

$$\phi_{i,j} = \frac{1}{\sqrt{8}}\left(1 + \frac{m_i}{m_j}\right)^{-1/2}\left[1 + \left(\frac{\eta_i}{\eta_j}\right)^{-1/2}\left(\frac{m_i}{m_j}\right)^{-1/4}\right]^2$$

Para t=0, el calor es cero.

Balance total de calor:

$$c_{v,mix}\frac{dT}{dt} = \frac{dQ}{dt} - P_i\frac{dV}{dt} + (h_w - U_w)\frac{dN_w}{dt}$$

$$c_{v,mix} = \sum_i (c_{v,i}\, N_i)$$

$$c_{v,i} = N_i k\left(\frac{f_i}{2} + \sum \frac{\left(\frac{\theta_i}{T}\right)^2 \exp\left(\frac{\theta_i}{T}\right)}{\left(\exp\left(\frac{\theta_i}{T}\right) - 1\right)^2}\right)$$

$$U_w = N_w k T\left(3 + \sum_{i=1}^{3} \frac{\theta_i/T}{\exp\left(\theta_i/T\right) - 1}\right)$$

$$h_w = 4kT_0$$

Para t=0, la temperatura inicial en el interior de la burbuja.

Sistema de ecuaciones cinéticas de craqueo:

$$v_{destilados} = K_1 C_{saturados}$$

$$v_{saturados} = K_2 C_{aromaticos} - K_1 C_{saturados}$$

$$v_{aromaticos} = K_3 C_{resinas} - K_2 C_{aromitcos}$$

$$v_{resinas} = K_5(C_{asfaletnos\ solubles} + C_{asfaltenos\ mesofase}) - (K_3 + 2K_4)C_{resinas}$$

$$v_{asfaletnos\ solubles} = K_4 C_{resinas} - K_5 C_{asfaltenos\ mesofase}$$

$$v_{asfaltenos\ mesofase} = K_4 C_{resinas} - (K_5 + K_6)C_{asfaltenos\ mesofase}$$

$$v_{coque} = K_6 C_{coque}$$

$$K_1 = \frac{1}{3600}\exp(29{,}9 - \frac{175000}{RT})$$

$$K_2 = \frac{1}{3600}\exp(23{,}43 - \frac{136000}{RT})$$

$$K_3 = \frac{1}{3600}\exp(13{,}57 - \frac{85000}{RT})$$

$$K_4 = \frac{1}{3600}\exp(17{,}07 - \frac{96000}{RT})$$

$$K_5 = \frac{1}{3600}\exp(19{,}34 - \frac{103000}{RT})$$

$$K_6 = \frac{1}{3600}\exp(27{,}54 - \frac{168000}{RT})$$

Método de Runge-Kutta-Felherger:

Forma general para la fórmula de Runge-Kutta de orden 4-5:

$$k_1 = hf(x_n, y_n)$$

$$k_1 = hf(x_n + a_2 h, y_n + b_{21} k_1)$$

…

$$k_6 = hf(x_n + a_6 h, y_n + b_{61} k_1 + \cdots + b_{65} k_5)$$

Fórmula para orden 5:

$$y_{n+1} = y_n + c_1k_1 + c_2k_2 + c_3k_3 + c_4k_4 + c_5k_5 + c_6k_6 + O(h^6)$$

Fórmula para orden 4:

$$y^*_{n+1} = y^*_n + c^*_1k_1 + c^*_1k_2 + c^*_2k_3 + c^*_4k_4 + c^*_5k_5 + c^*_6k_6 + O(h^5)$$

Cálculo del paso

$$h_0 \begin{cases} sh_1 \left|\frac{\Delta_0}{\Delta_1}\right|^{0.2} & \Delta_0 \geq \Delta_1 \\ sh_1 \left|\frac{\Delta_0}{\Delta_1}\right|^{0.25} & \Delta_0 < \Delta_1 \end{cases}$$

Error estimado

$$\Delta_1 = y_{n+1} - y^*_{n+1}$$

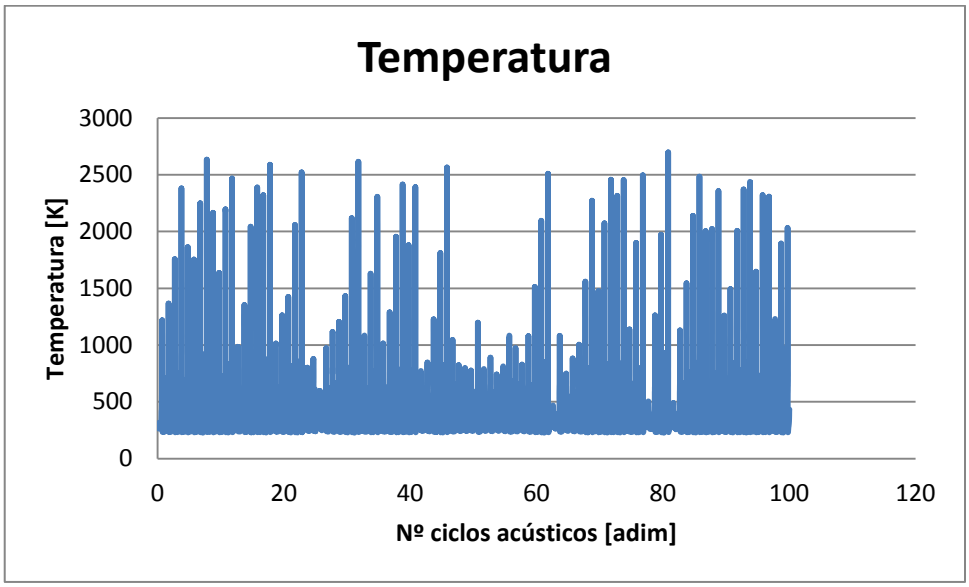
$$\Delta_0 = \epsilon y_{scal}$$

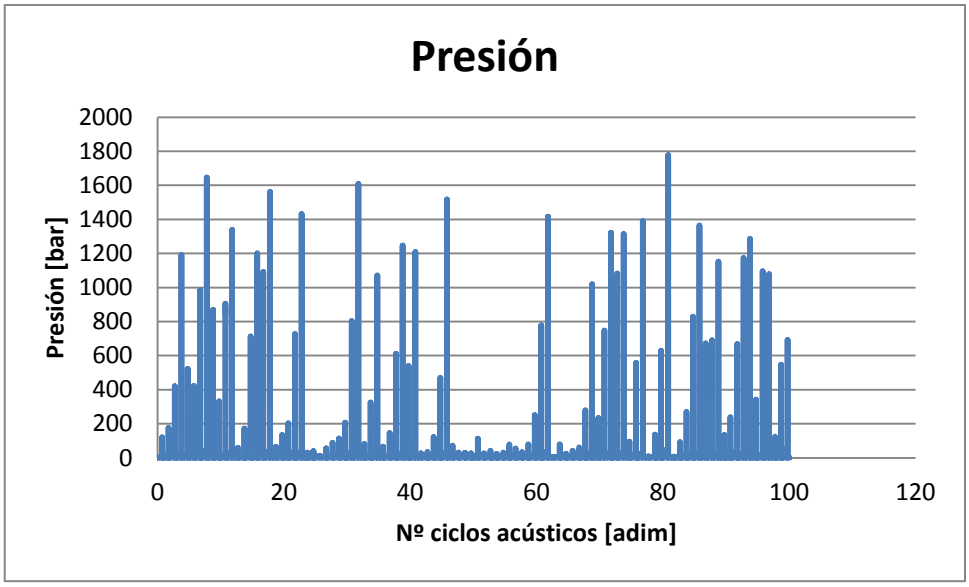
$$y_{scal} = y_n + k_1$$

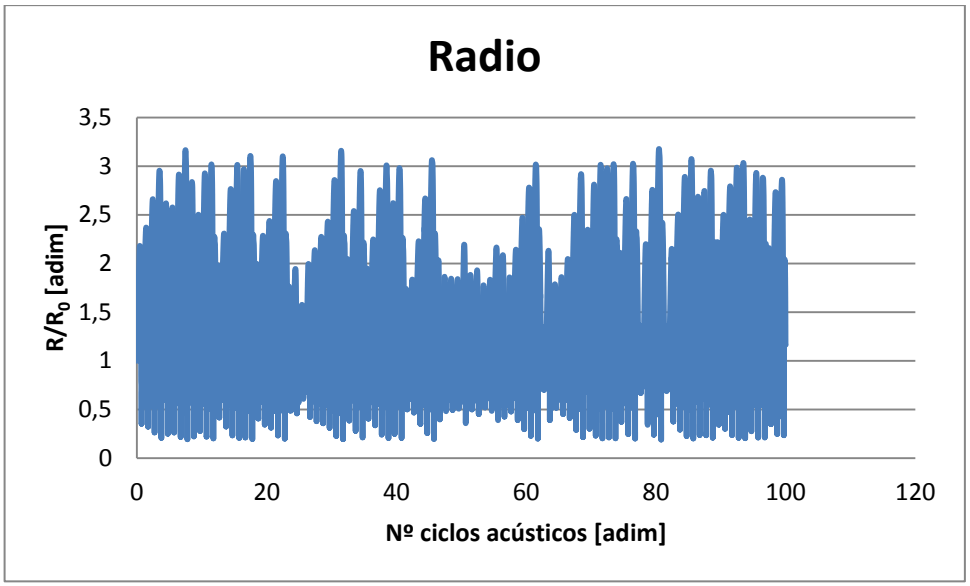
i	a_i	b_{ij}					c_i	c_i^*
1							37/378	2825/27648
2	1/5	1/5					0	0
3	3/10	3/40	9/40				250/621	18575/48384
4	3/5	3/10	-9/10	6/5			125/594	13525/55296
5	1	-11/54	5/2	-70/27	35/27		0	277/141336
6	7/8	1631/55296	175/512	575/13824	44275/110592	253/4096	512/1771	1/4
j=		1	2	3	4	5		

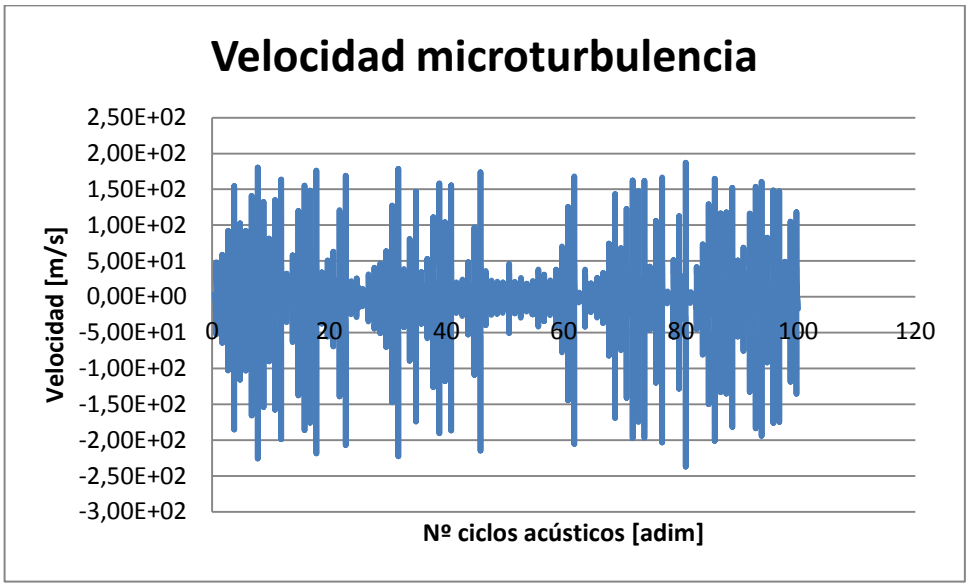
ANEXO V

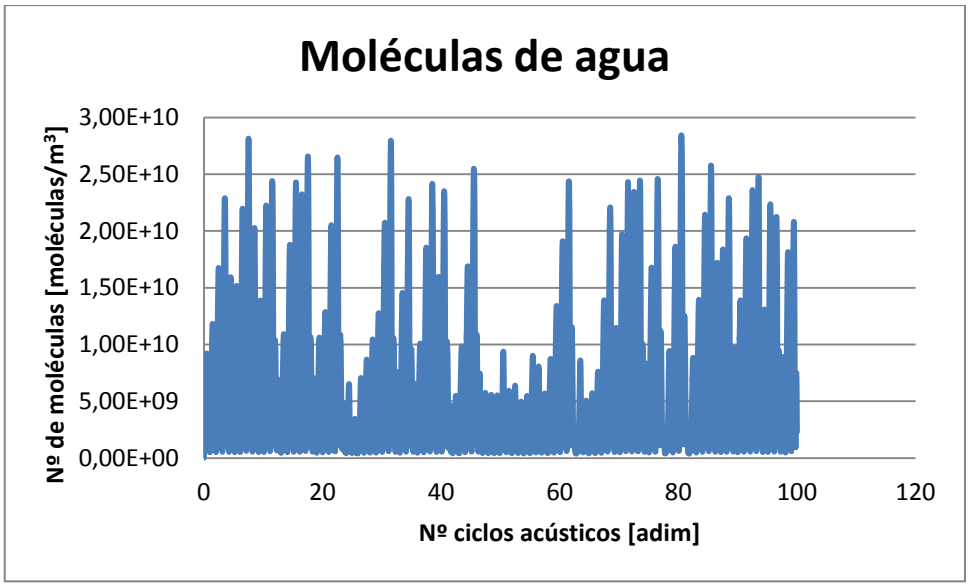
En este anexo se adjuntan los resultados del programa de simulación numérica para el caso base:







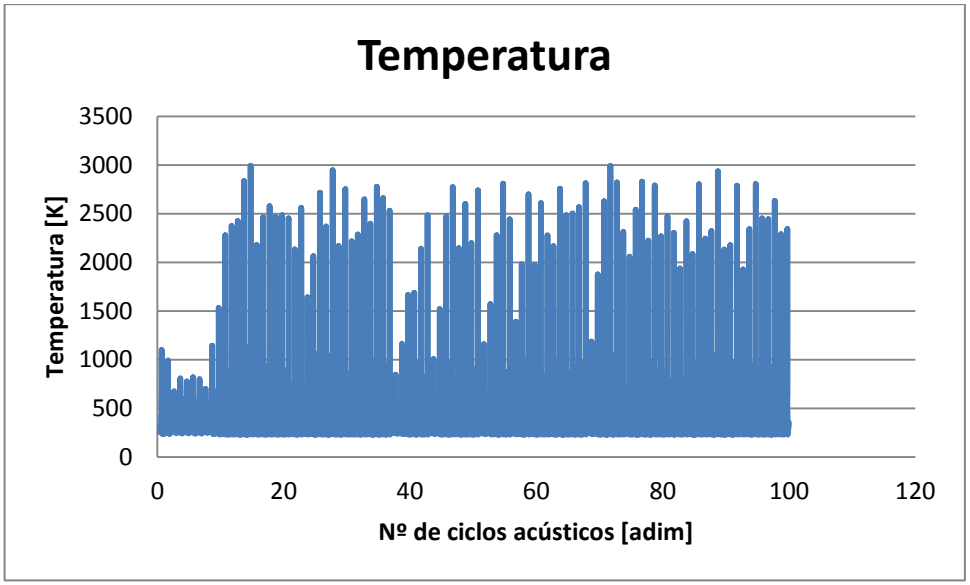


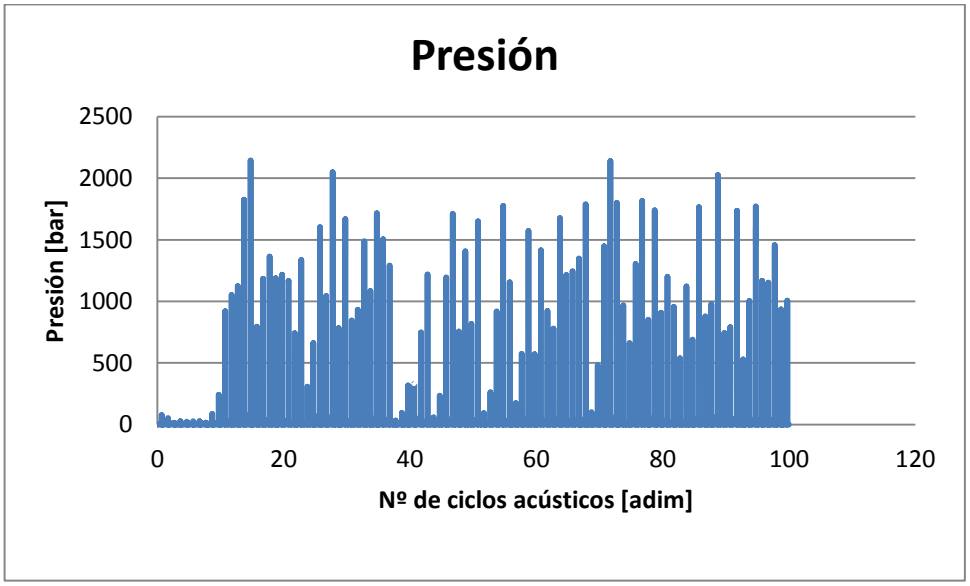

Moléculas de agua
Nº de moléculas [moléculas/m³]
3,00E+10
2,50E+10
2,00E+10
1,50E+10
1,00E+10
5,00E+09
0,00E+00
0
20
40
60
80
100
120
Nº ciclos acústicos [adim]

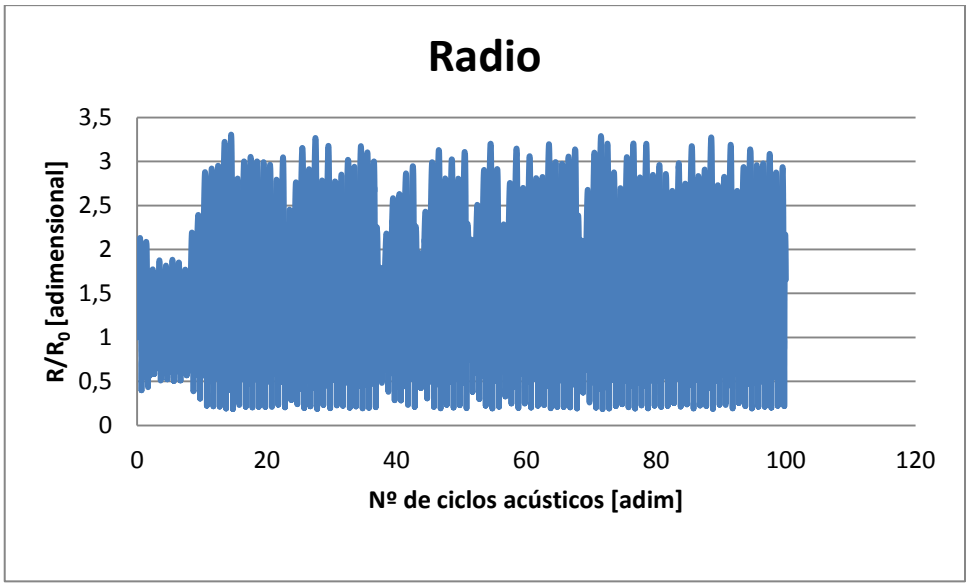
Anexo VI

En este anexo se muestran las gráficas de picos de presión, temperatura y evolución del radio de burbuja del análisis paramétrico de temperatura inicial del líquido. Las gráficas siguientes son para T = 283,15 K, T=323,15 K y T=383,15 K, y para el intervalo de puntos que no siguen la evolución de concentraciones de productos finales reseñada en el correspondiente estudio. Se puede comprobar que para esos puntos anómalos se cumple lo que dicta la teoría de cavitación. Para el resto de temperaturas, sus respectivas gráficas son situaciones intermedias de las que se muestran y, por tanto, no se muestran.

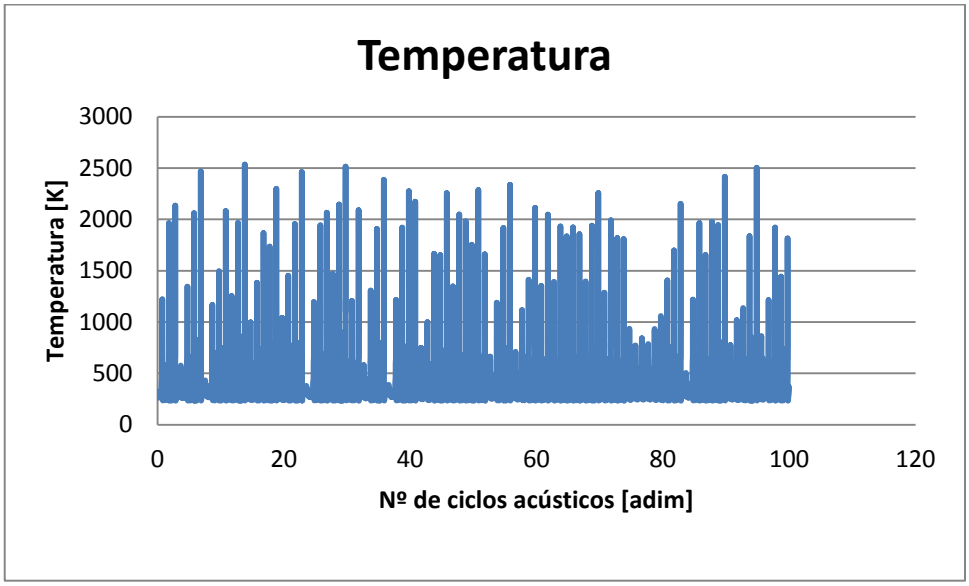
T = 283,15 K:

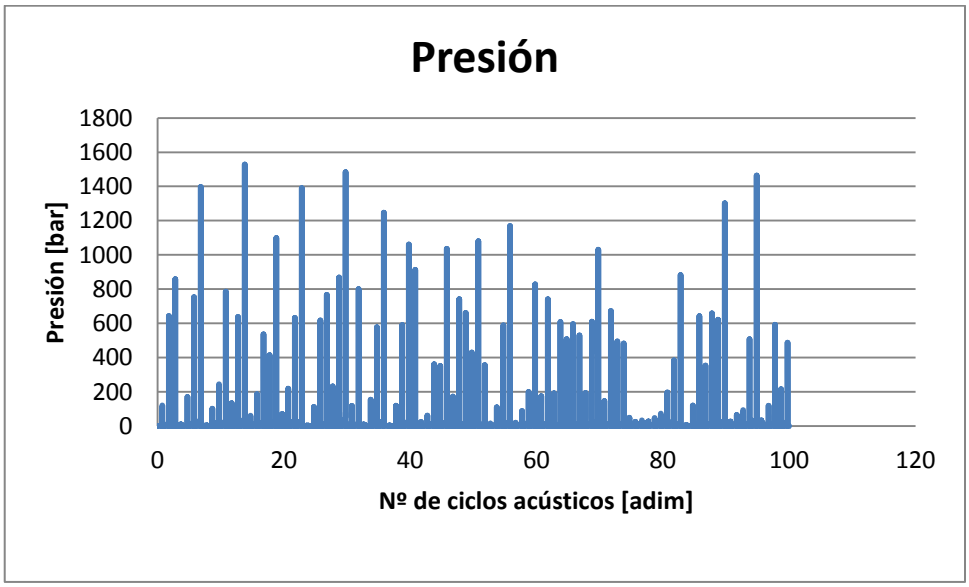


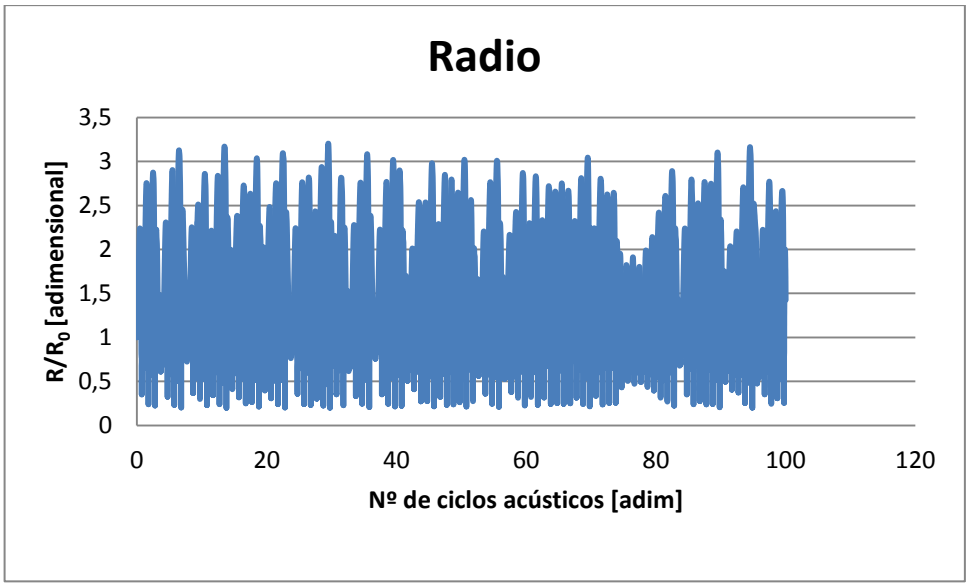



Radio
3,5
3
2,5
2
1,5
1
0,5
0
R/R0 [adimensional]
0
20
40
60
80
100
120
Nº de ciclos acústicos [adim]

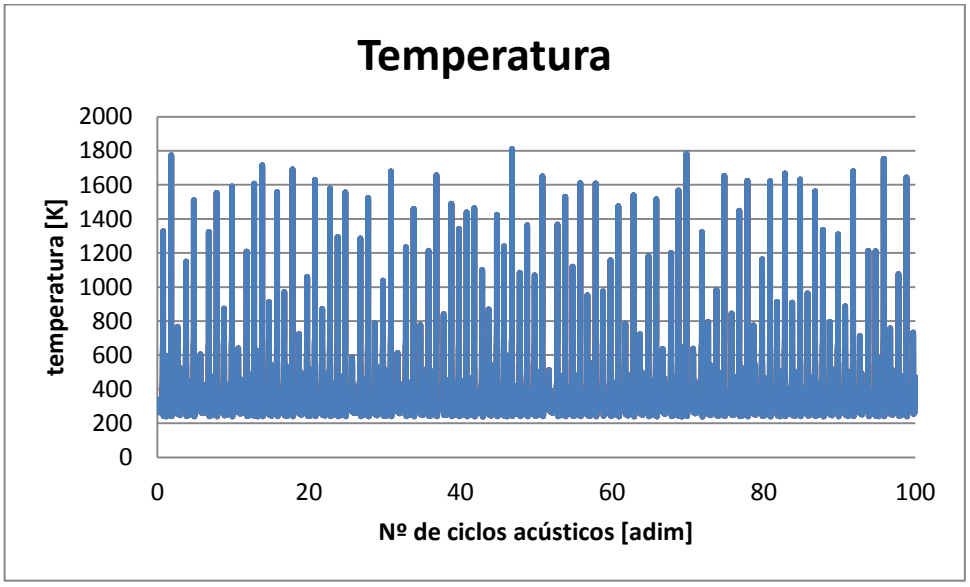
T = 323,15 K

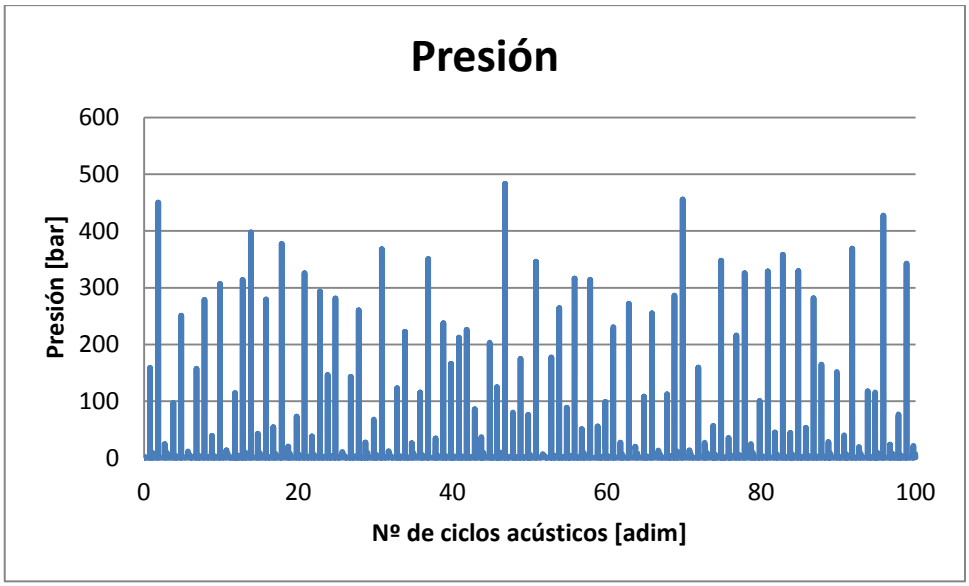

Temperatura
3000
2500
2000
1500
1000
500
0
Temperatura [K]
0
20
40
60
80
100
120
Nº de ciclos acústicos [adim]

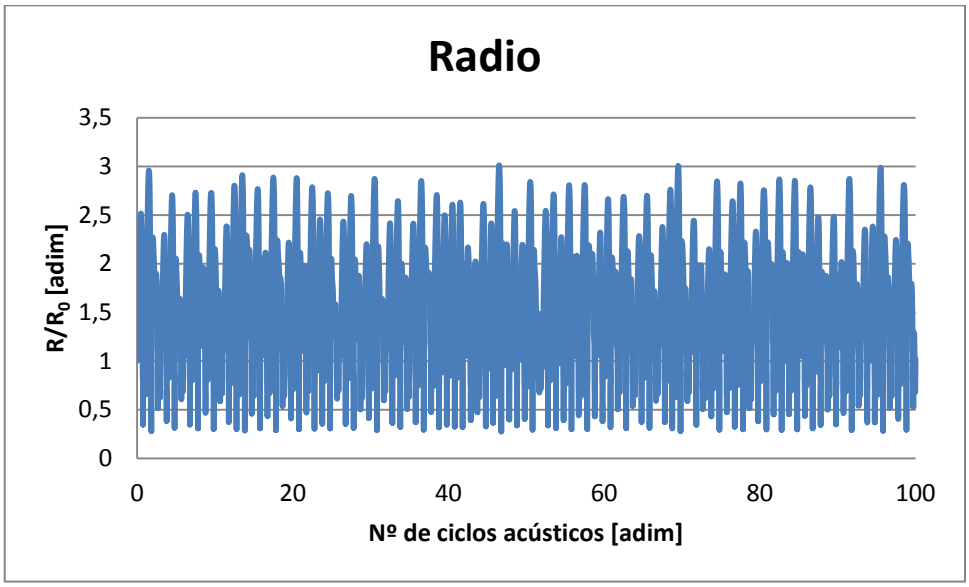

Presión
1800
1600
1400
1200
1000
800
600
400
200
0
Presión [bar]
0
20
40
60
80
100
120
Nº de ciclos acústicos [adim]



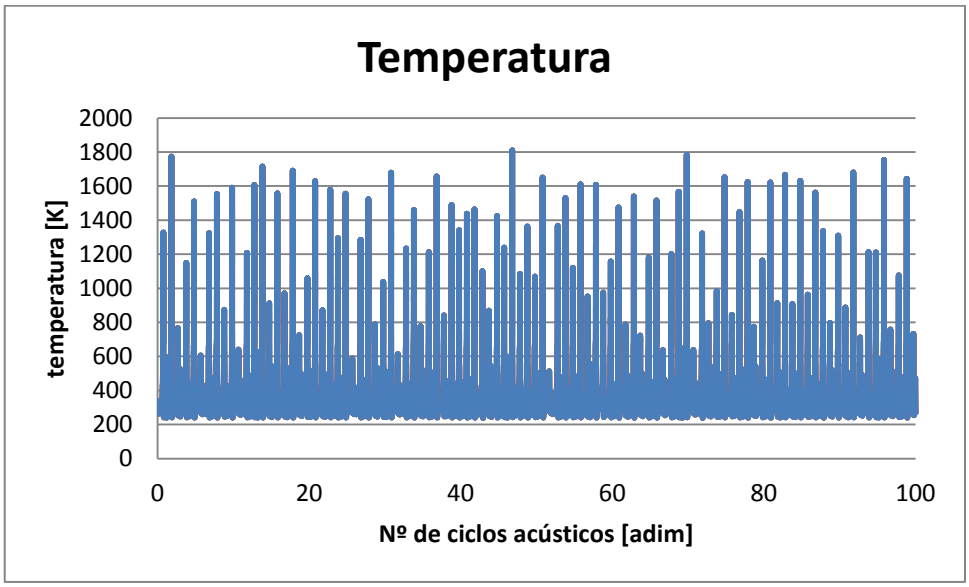
T = 338,15 K

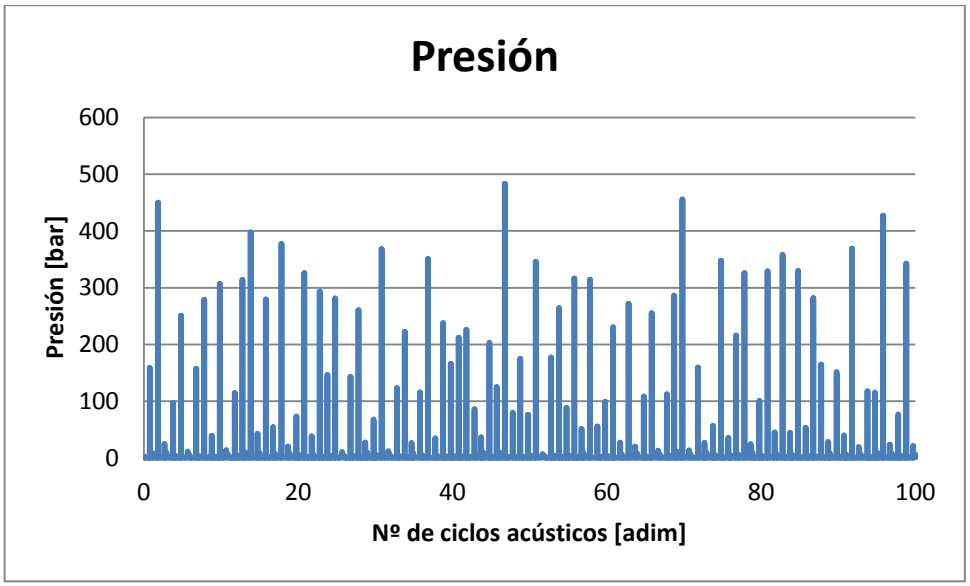


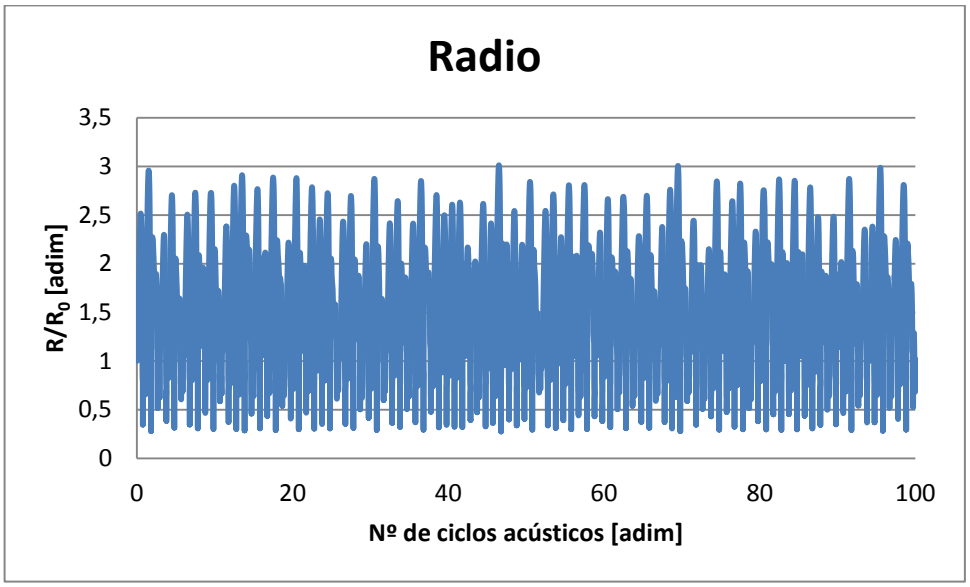

Presión
600
500
400
300
200
100
0
Presión [bar]
0
20
40
60
80
100
Nº de ciclos acústicos [adim]


Radio
3,5
3
2,5
2
1,5
1
0,5
0
R/R0 [adim]
0
20
40
60
80
100
Nº de ciclos acústicos [adim]

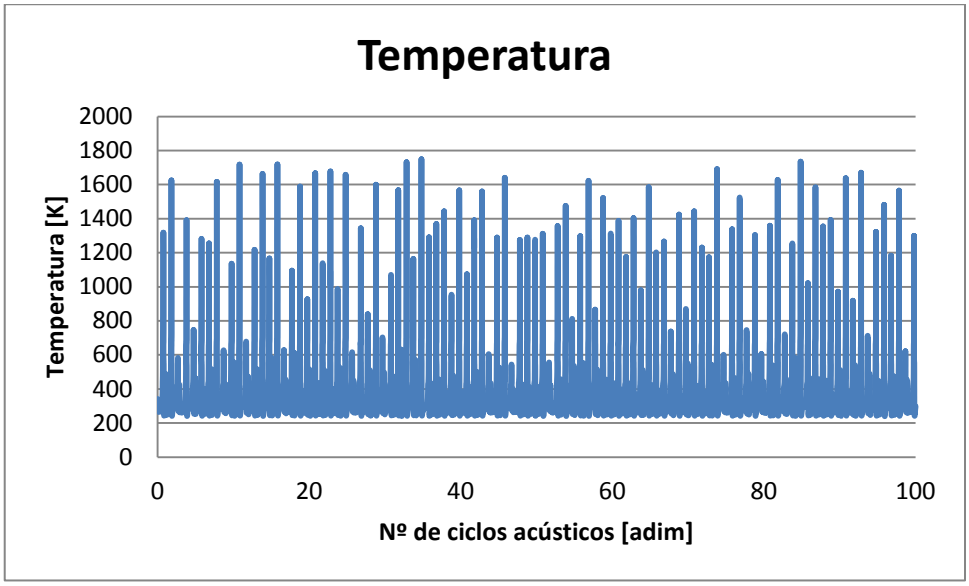
T = 340,15 K

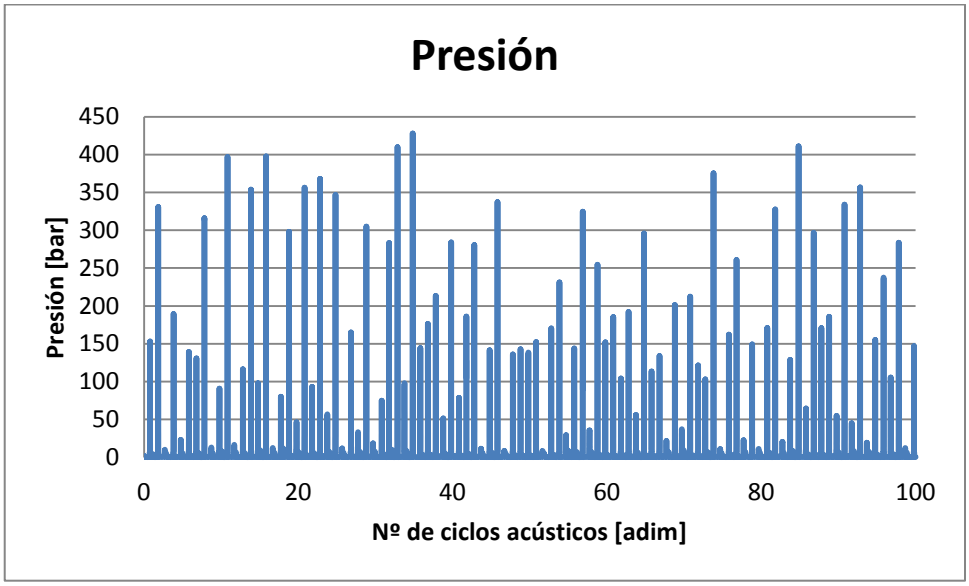

Temperatura
2000
1800
1600
1400
1200
1000
800
600
400
200
0
temperatura [K]
0
20
40
60
80
100
Nº de ciclos acústicos [adim]

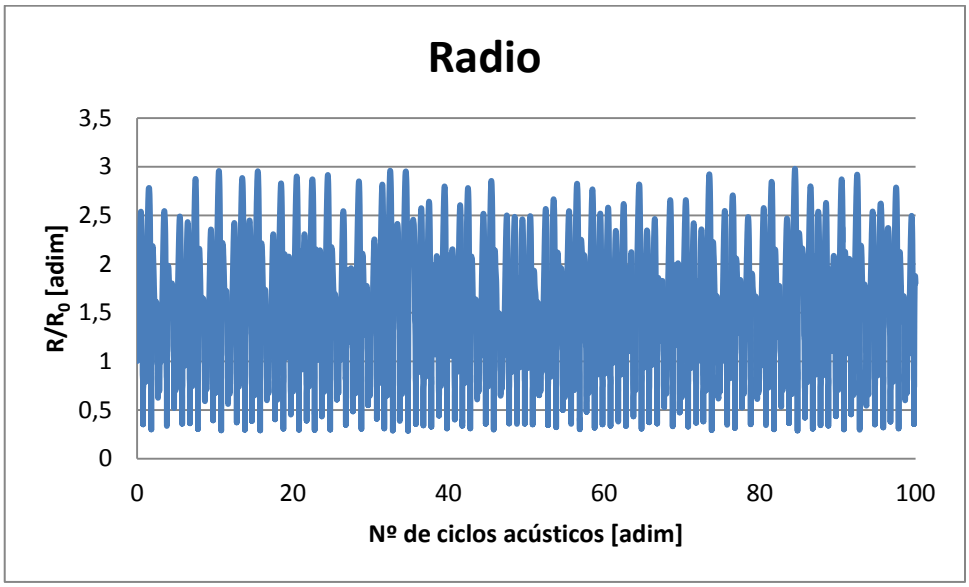




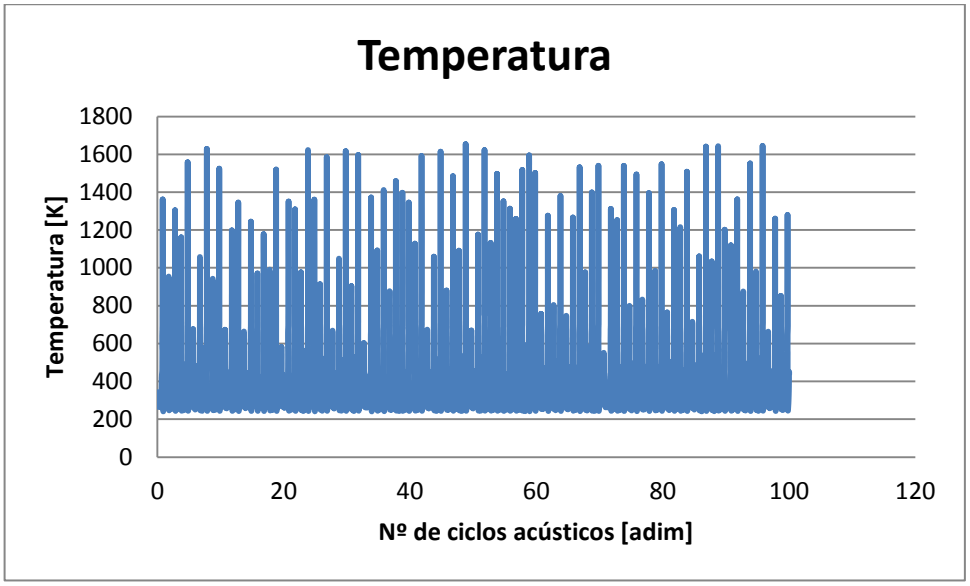
T = 341,15 K

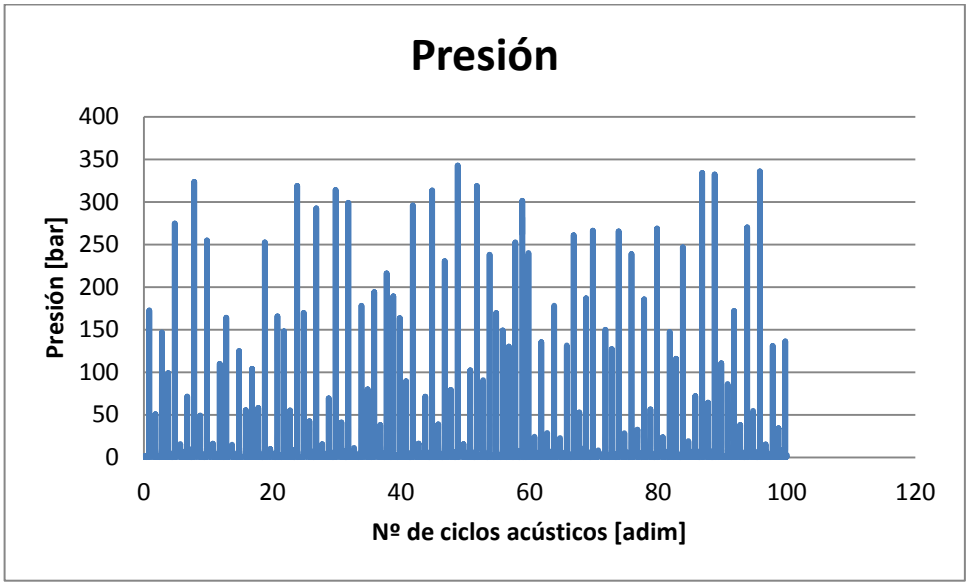

Temperatura
Temperatura [K]
2000
1800
1600
1400
1200
1000
800
600
400
200
0
0
20
40
60
80
100
Nº de ciclos acústicos [adim]

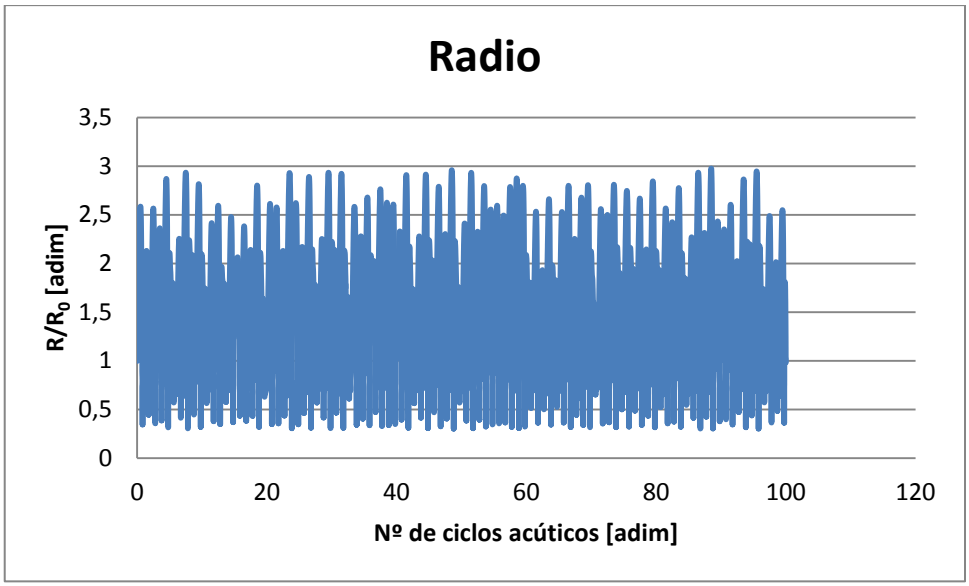

Presión
Presión [bar]
450
400
350
300
250
200
150
100
50
0
0
20
40
60
80
100
Nº de ciclos acústicos [adim]


Radio
R/R0 [adim]
3,5
3
2,5
2
1,5
1
0,5
0
0
20
40
60
80
100
Nº de ciclos acústicos [adim]

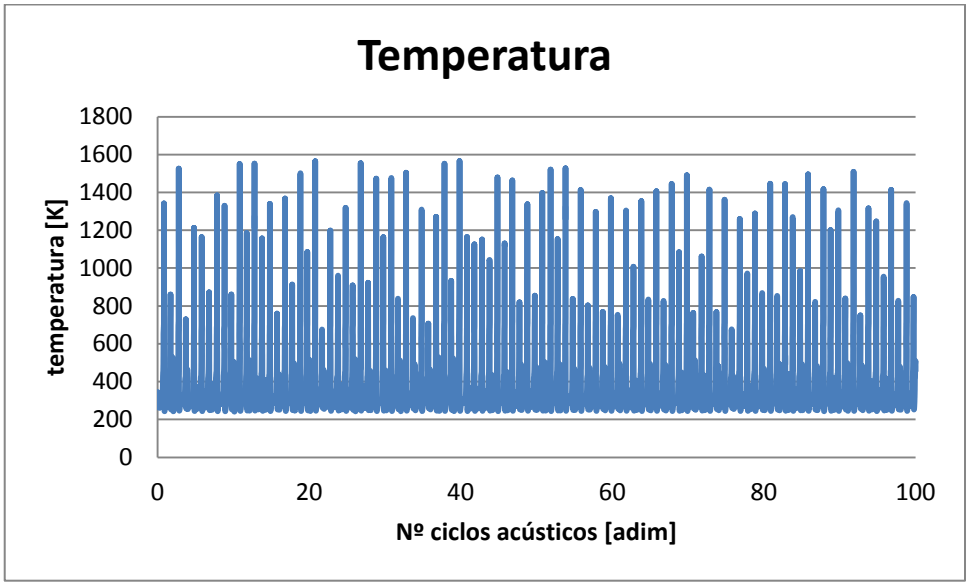
T = 343,15 K

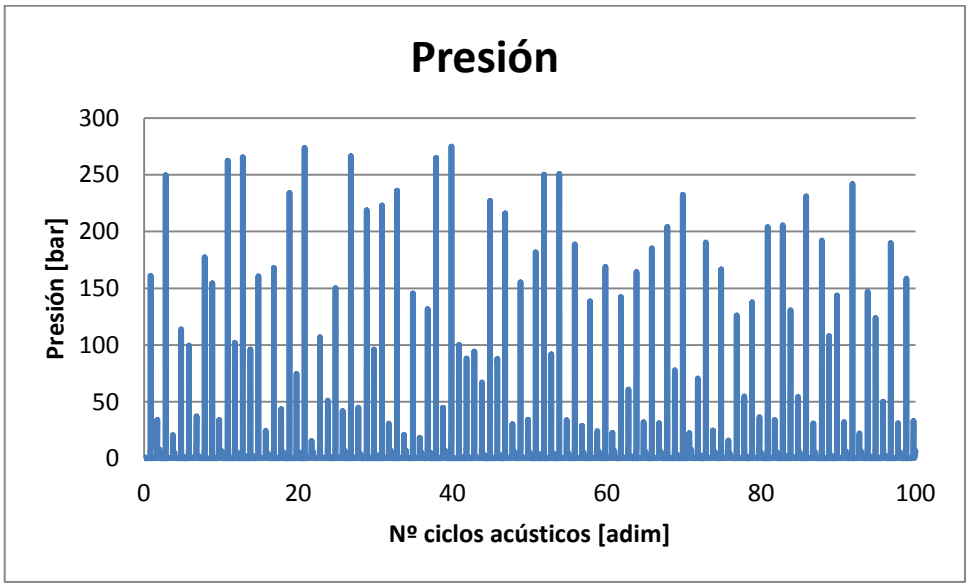


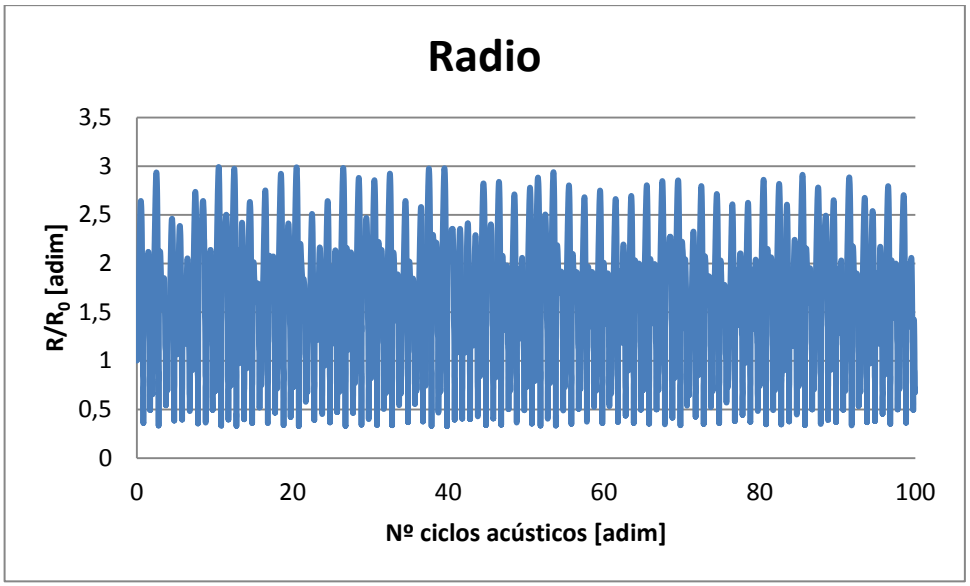



Radio
3,5
3
2,5
2
1,5
1
0,5
0
R/R0 [adim]
0
20
40
60
80
100
120
Nº de ciclos acúticos [adim]

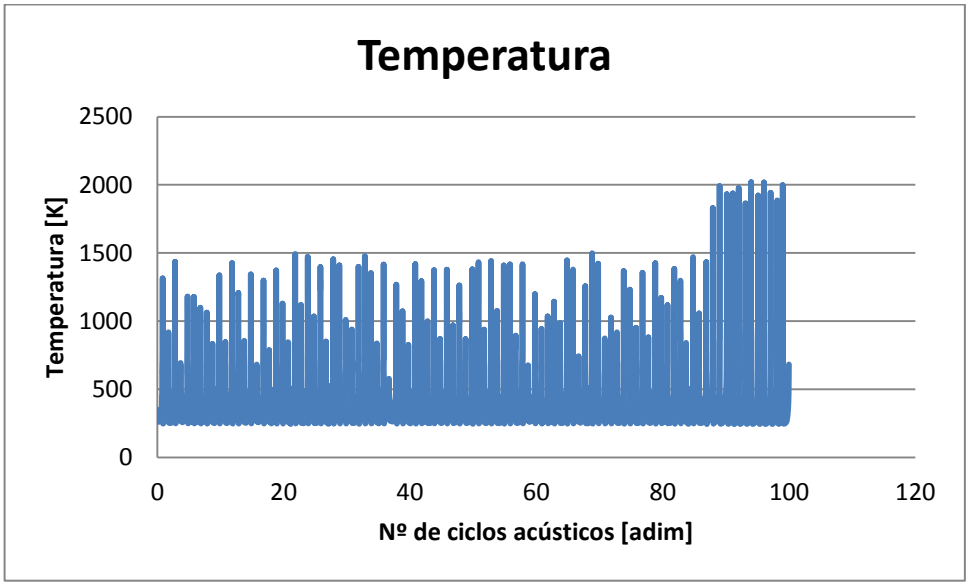
T = 346,15 K

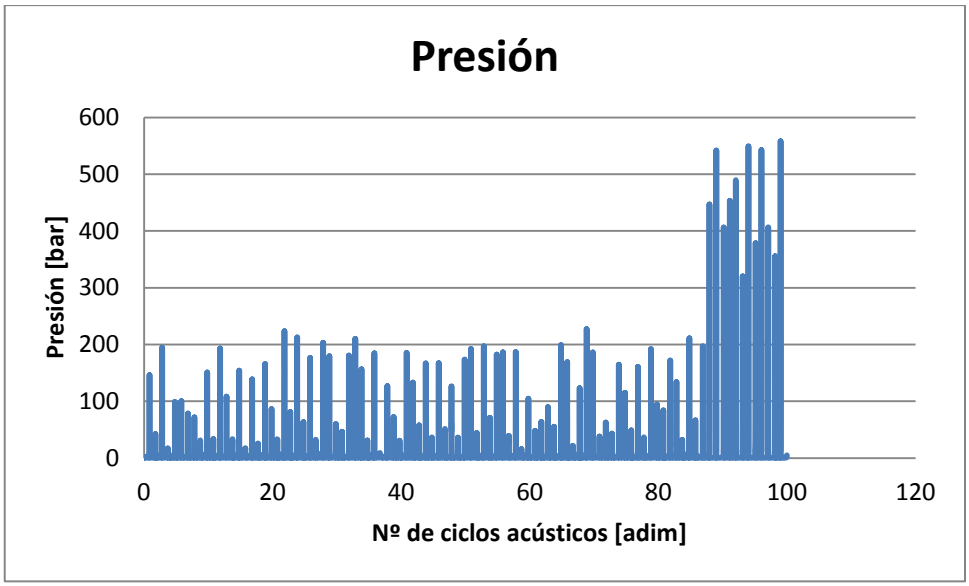

Temperatura
1800
1600
1400
1200
1000
800
600
400
200
0
temperatura [K]
0
20
40
60
80
100
Nº ciclos acústicos [adim]

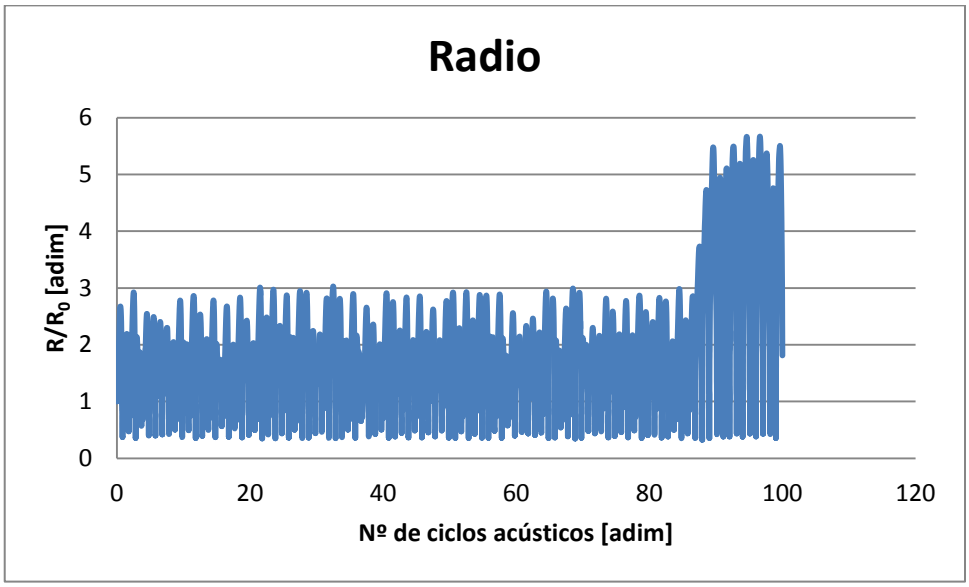

Presión
300
250
200
150
100
50
0
Presión [bar]
0
20
40
60
80
100
Nº ciclos acústicos [adim]



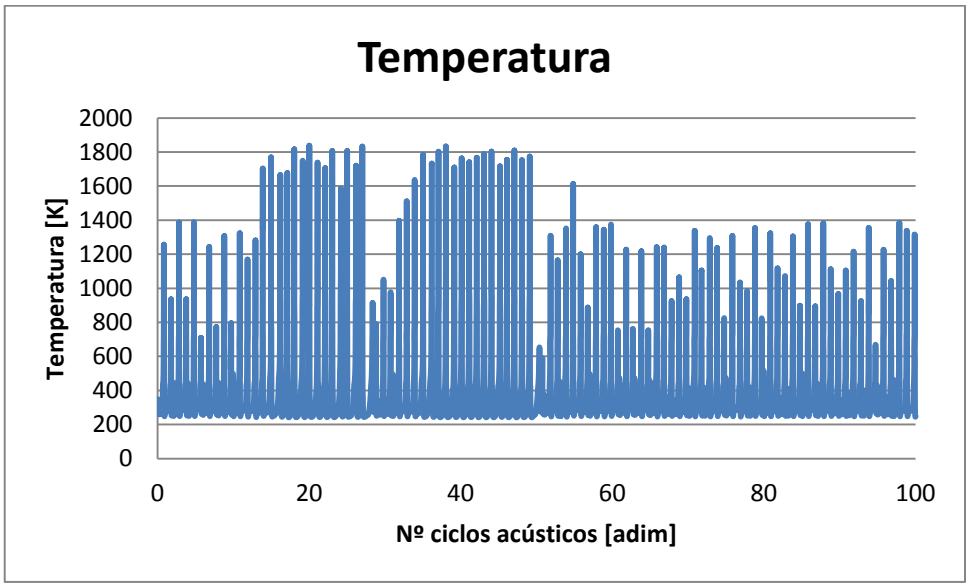
T = 348,15 K

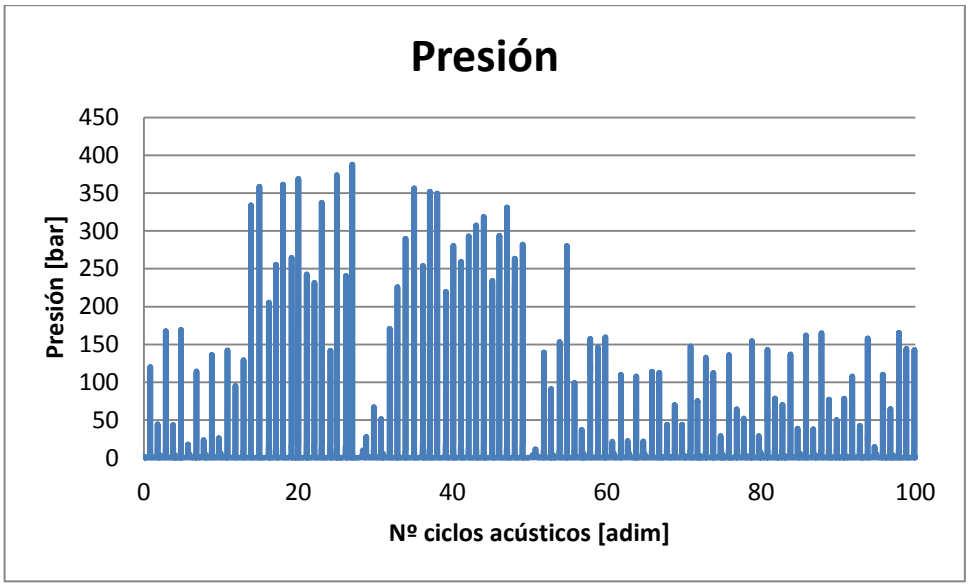


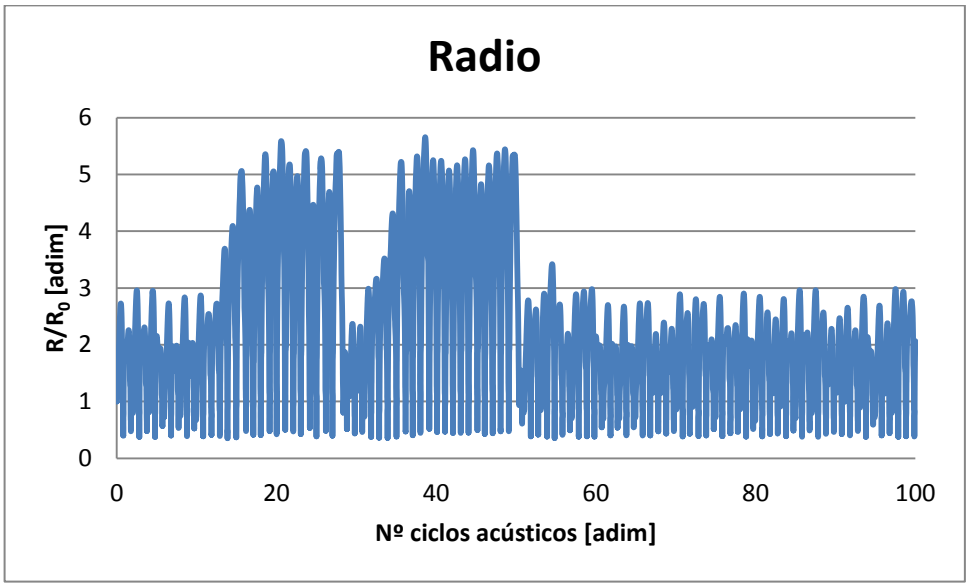

Presión
600
500
400
300
200
100
0
Presión [bar]
0
20
40
60
80
100
120
Nº de ciclos acústicos [adim]


Radio
6
5
4
3
2
1
0
R/R0 [adim]
0
20
40
60
80
100
120
Nº de ciclos acústicos [adim]

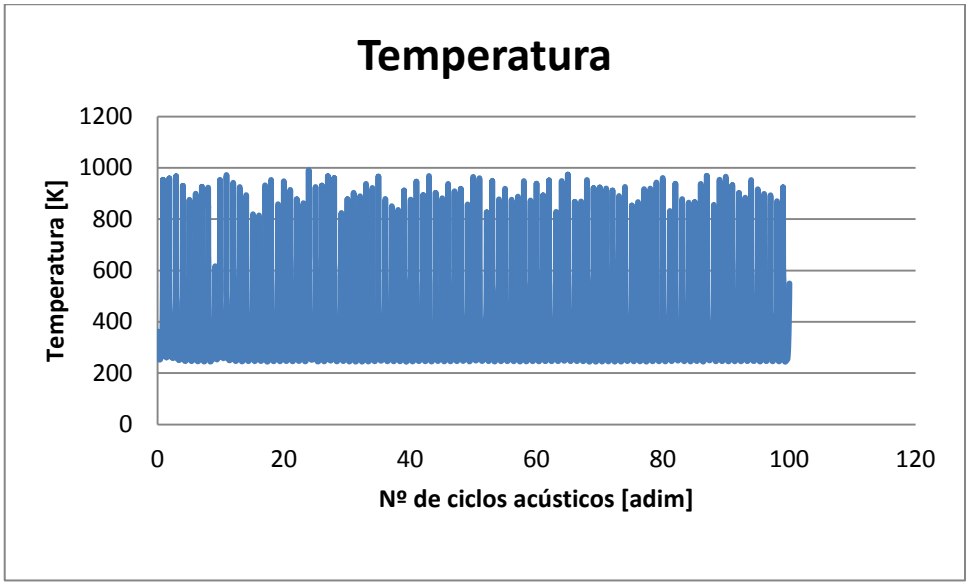
T = 350,15 K

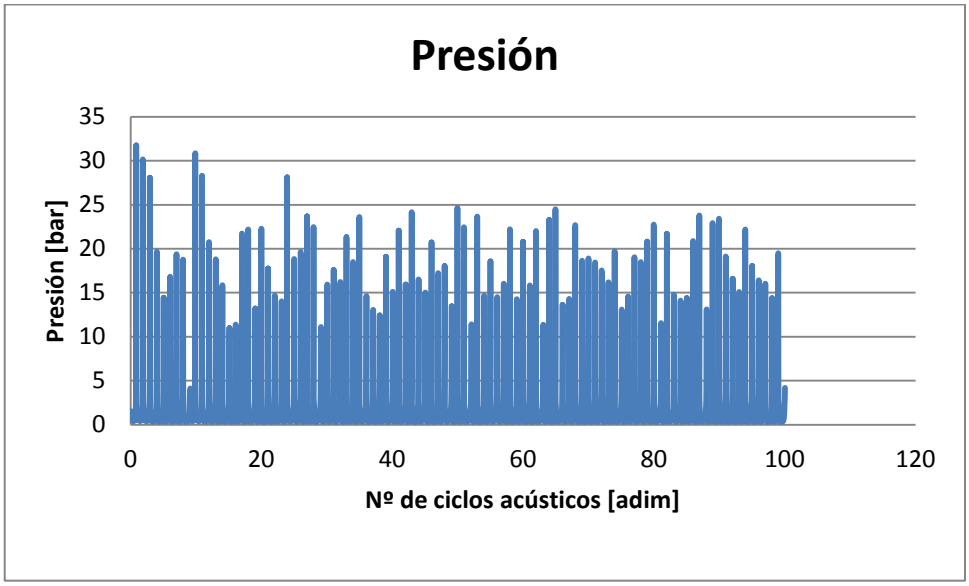

Temperatura
2000
1800
1600
1400
1200
1000
800
600
400
200
0
Temperatura [K]
0
20
40
60
80
100
Nº ciclos acústicos [adim]

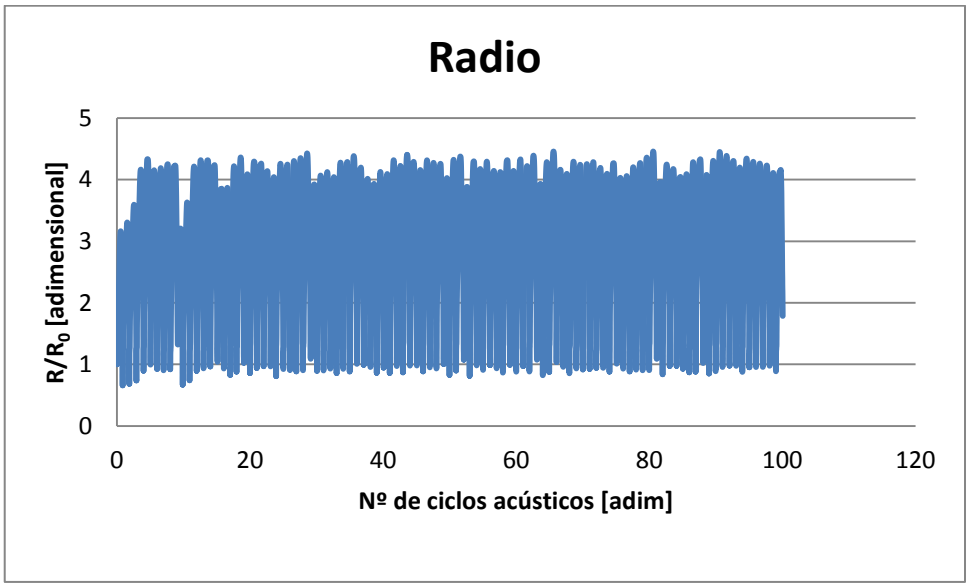




T = 383,15 K


Temperatura
Temperatura [K]
1200
1000
800
600
400
200
0
0
20
40
60
80
100
120
Nº de ciclos acústicos [adim]

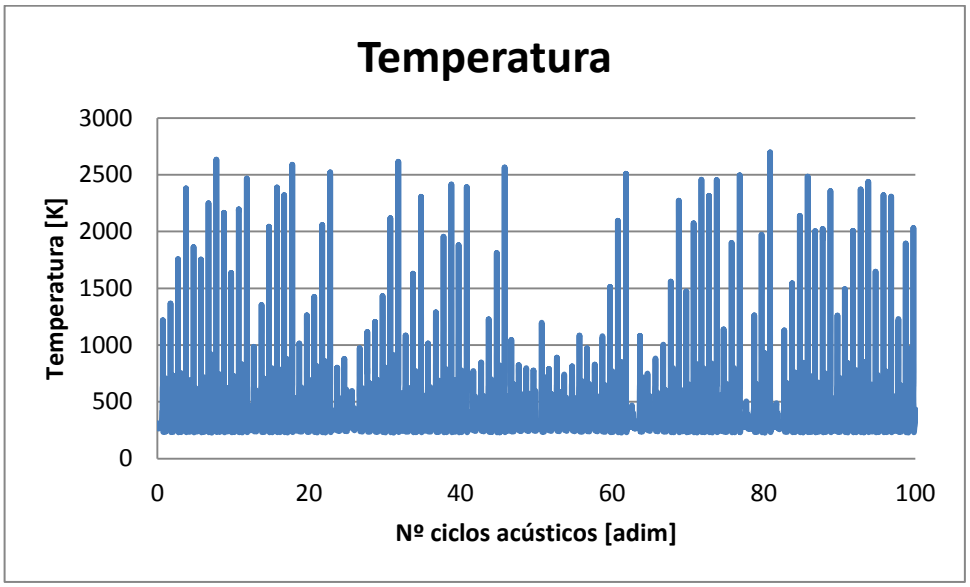

Presión
Presión [bar]
35
30
25
20
15
10
5
0
0
20
40
60
80
100
120
Nº de ciclos acústicos [adim]


Radio
R/R0 [adimensional]
5
4
3
2
1
0
0
20
40
60
80
100
120
Nº de ciclos acústicos [adim]

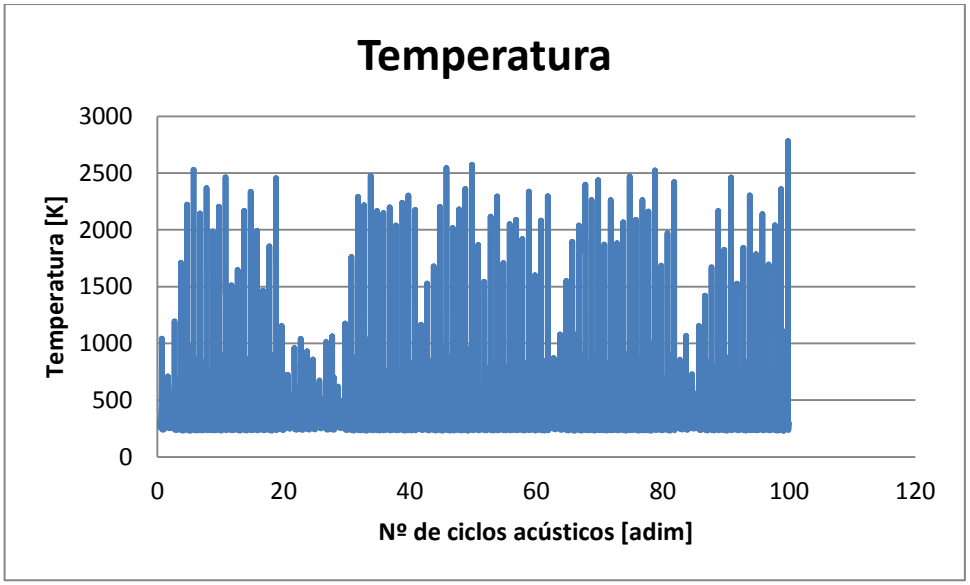
ANEXO VII

A continuación se muestran las gráficas de la evolución de temperatura de la burbuja para el estudio paramétrico de la presión inicial del líquido.

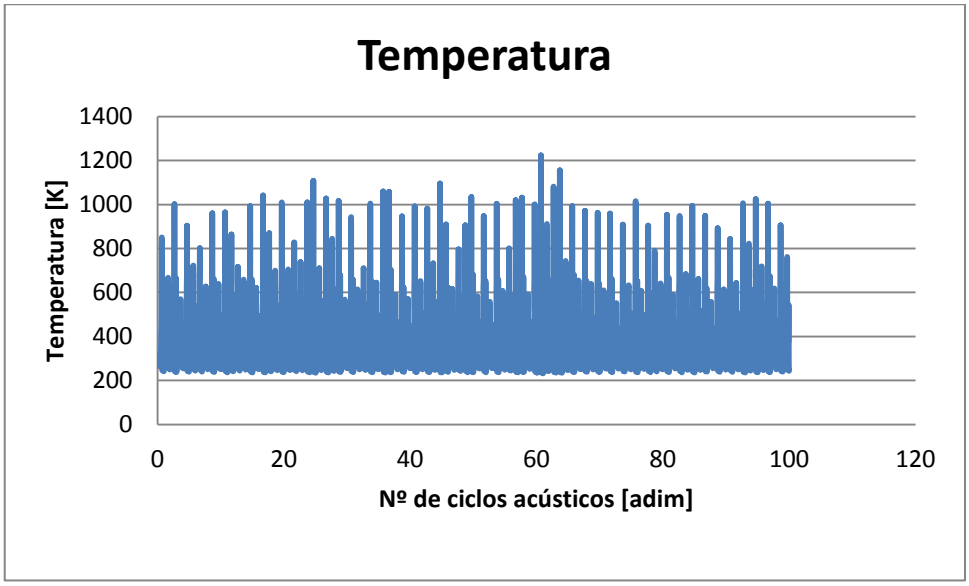
Presión inicial 1 bar:



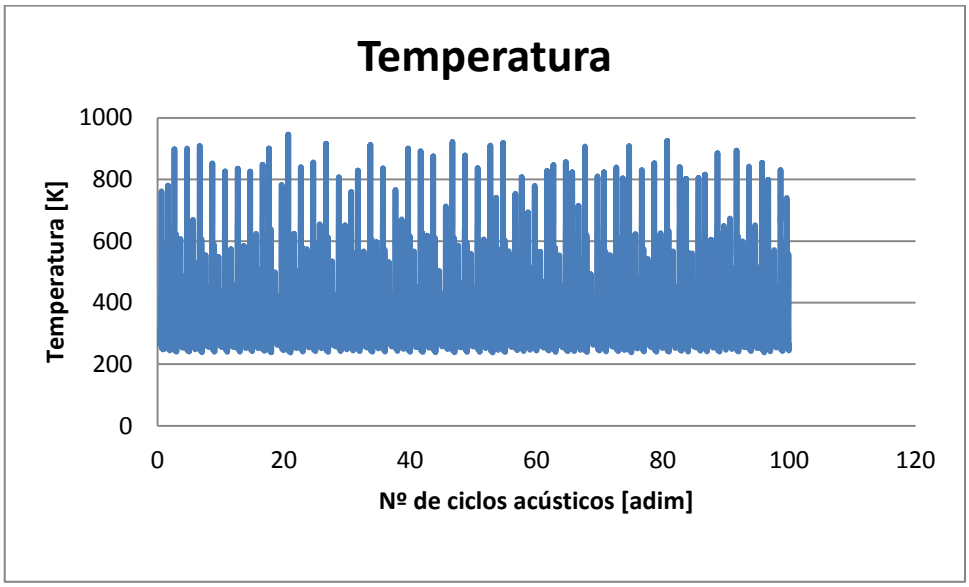
Presión inicial 1,05 bar:



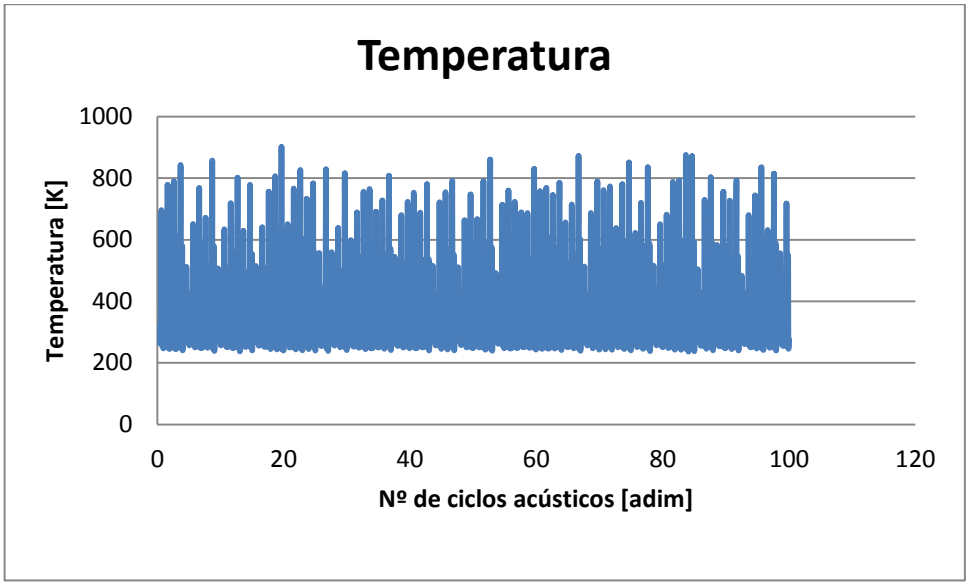
Presión inicial 1,10 bar:



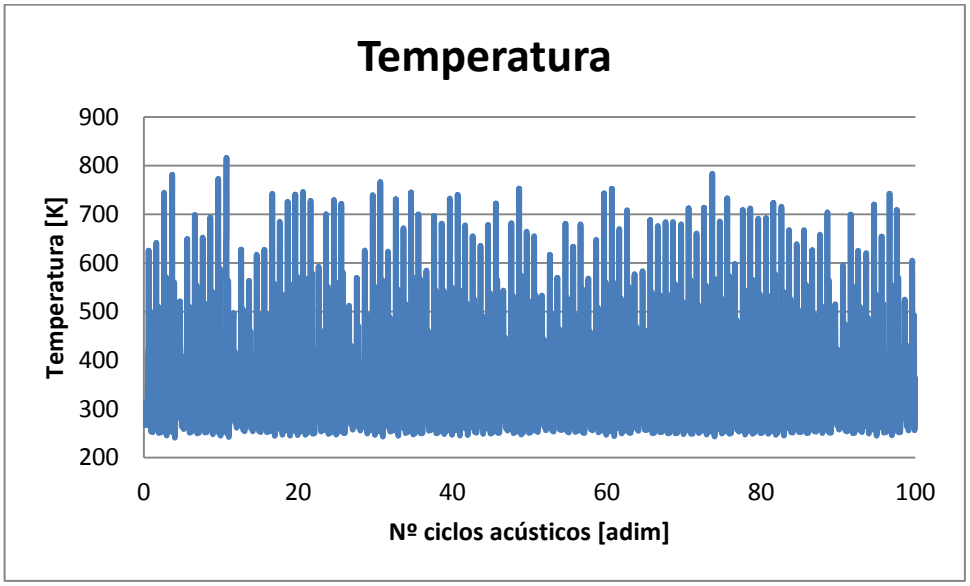
Presión inicial 1,15 bar:



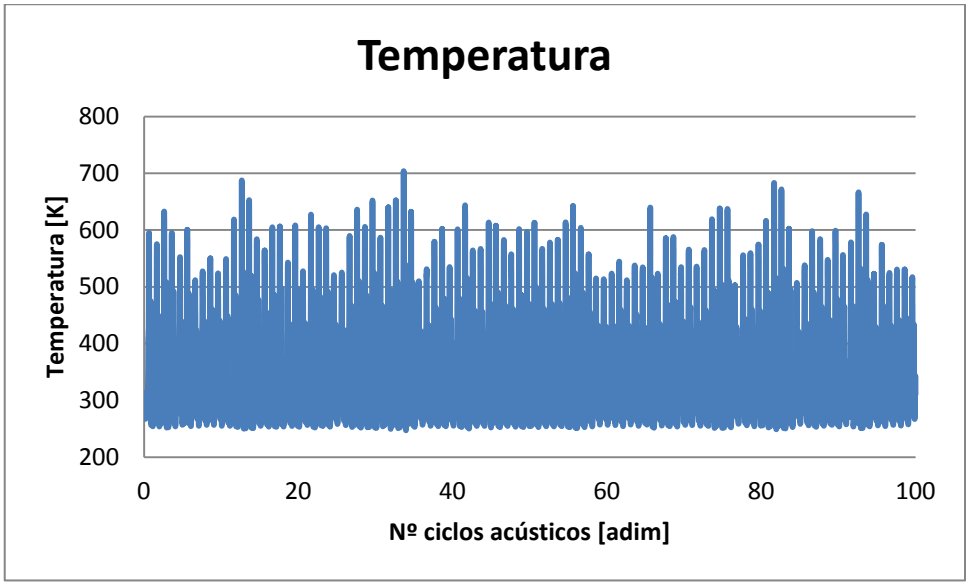
Presión inicial 1,20 bar:



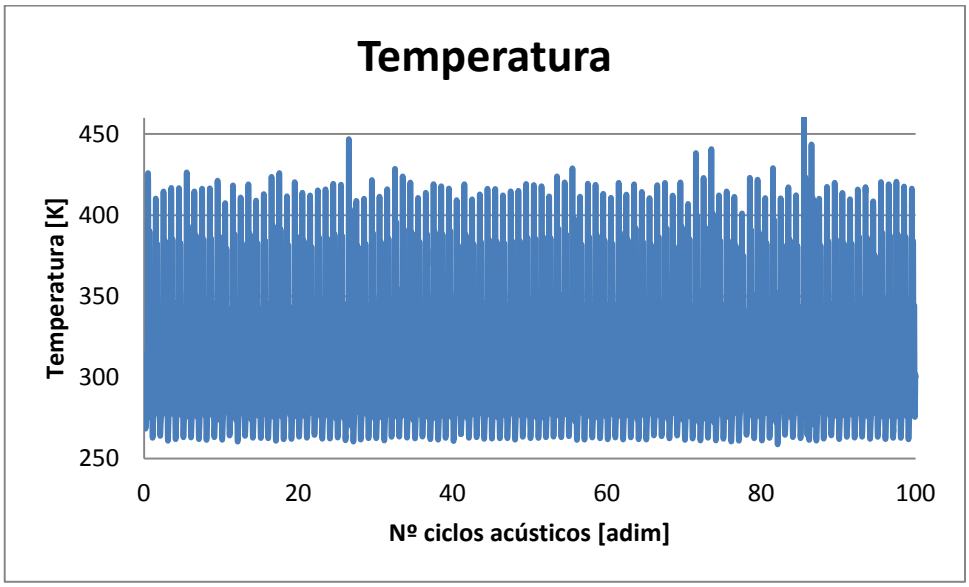
Presión inicial 1,25 bar:



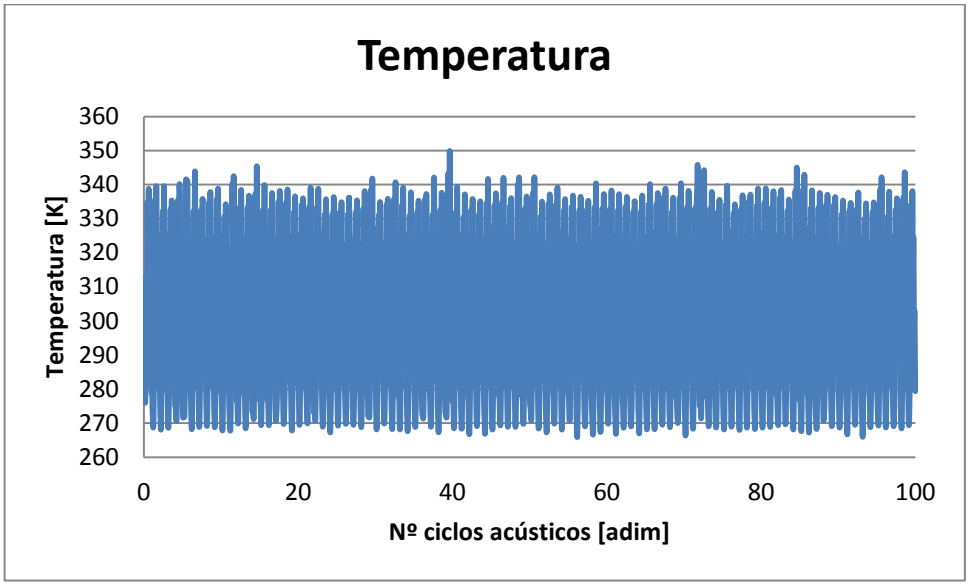
Presión inicial 1,3 bar:



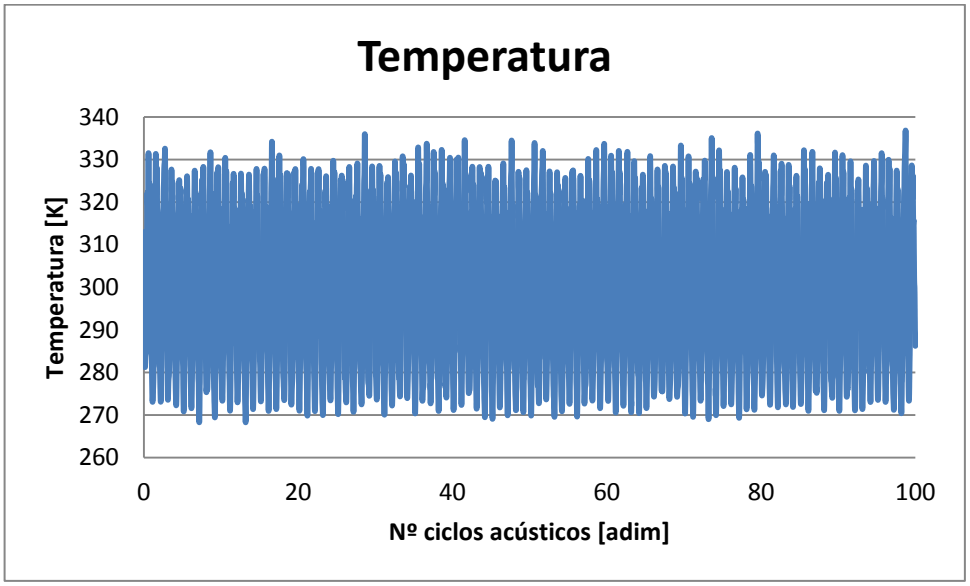
Presión inicial 1,5 bar:



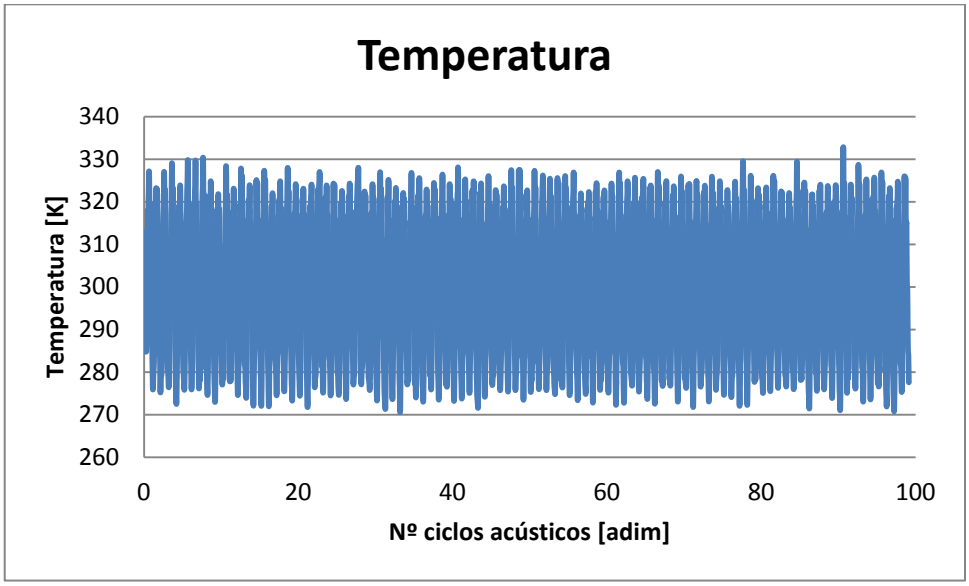
Presión inicial 2 bar:



Presión inicial 2,5 bar:



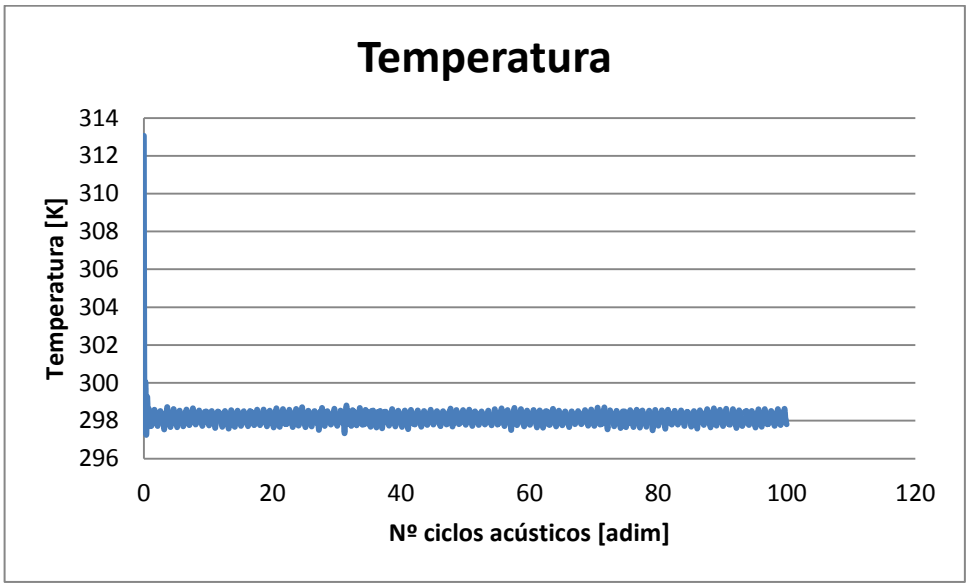
Presión inicial 3 bar:



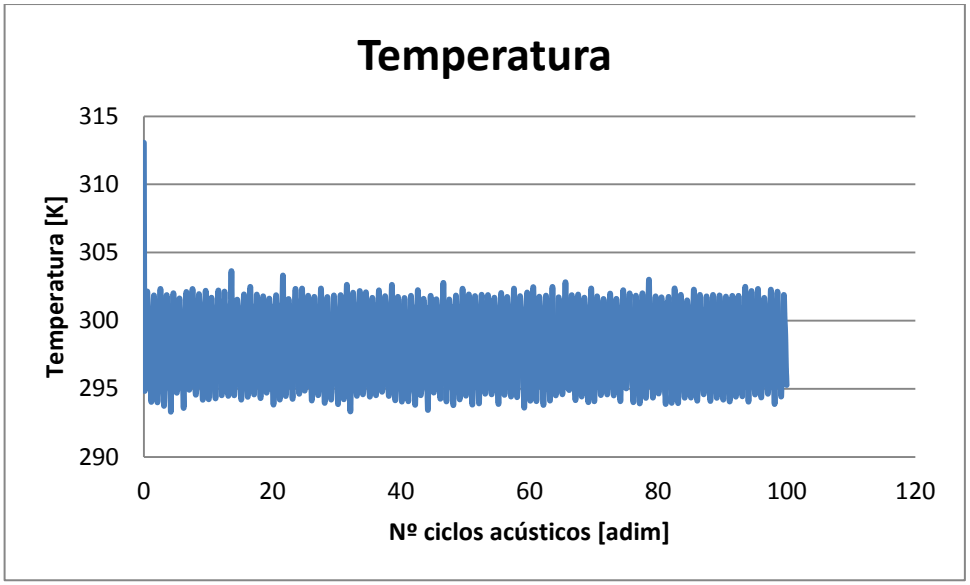
ANEXO VIII

A continuación se muestran las gráficas de la evolución de temperatura de la burbuja para el estudio paramétrico de la amplitud de ultrasonido.

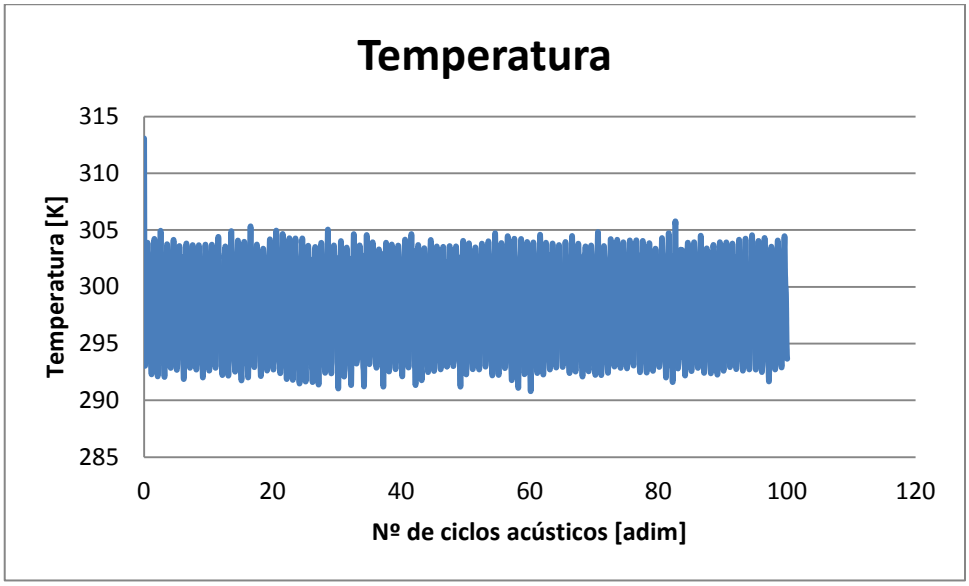
Amplitud de ultrasonido 0,01 bar:



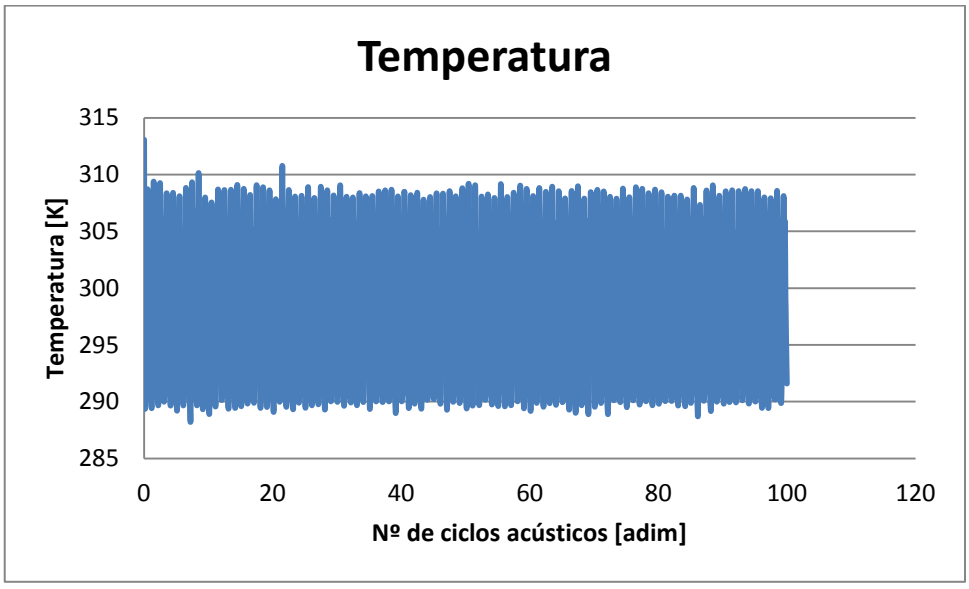
Amplitud de ultrasonido 0,1 bar:



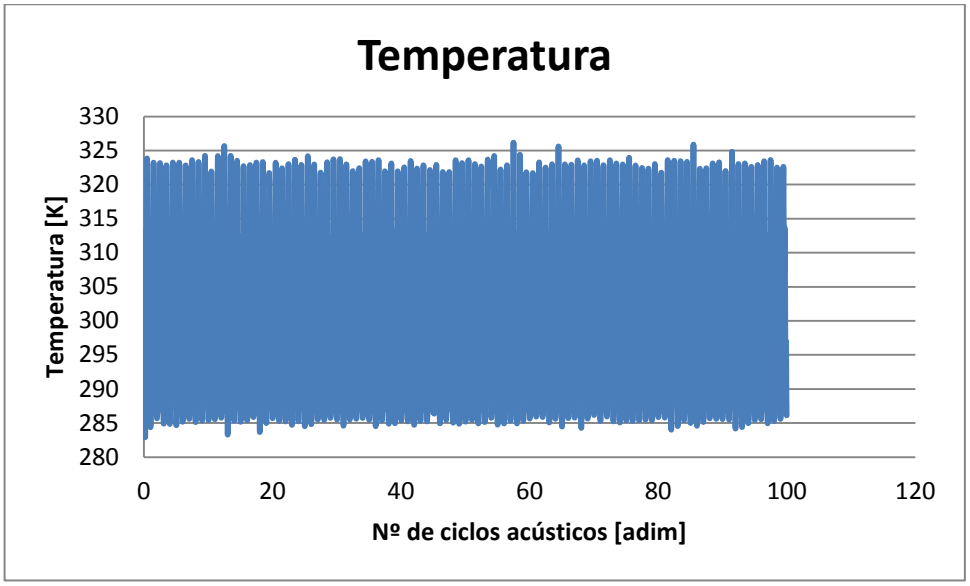
Amplitud de ultrasonido 0,15 bar:



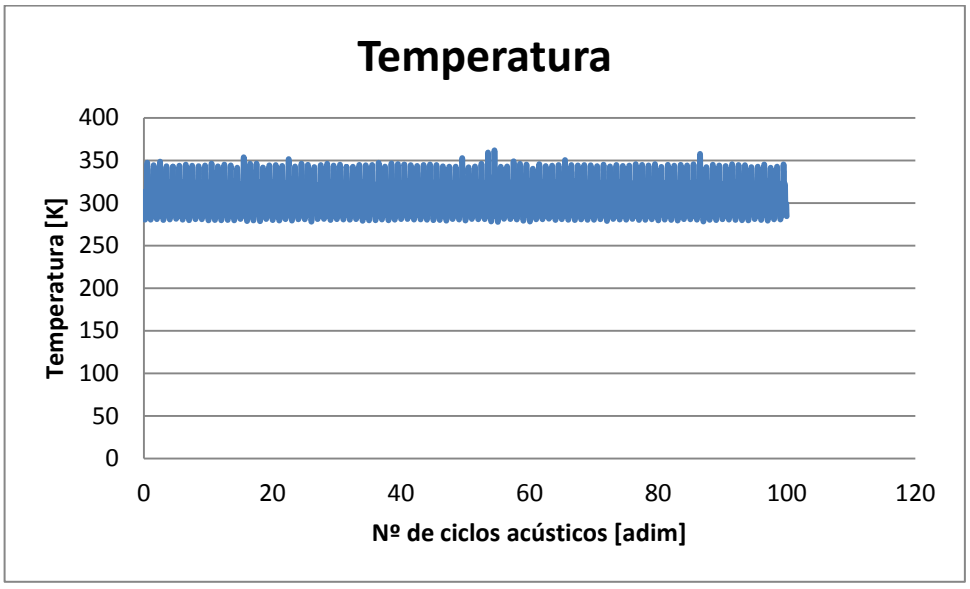
Amplitud de ultrasonido 0,25 bar:



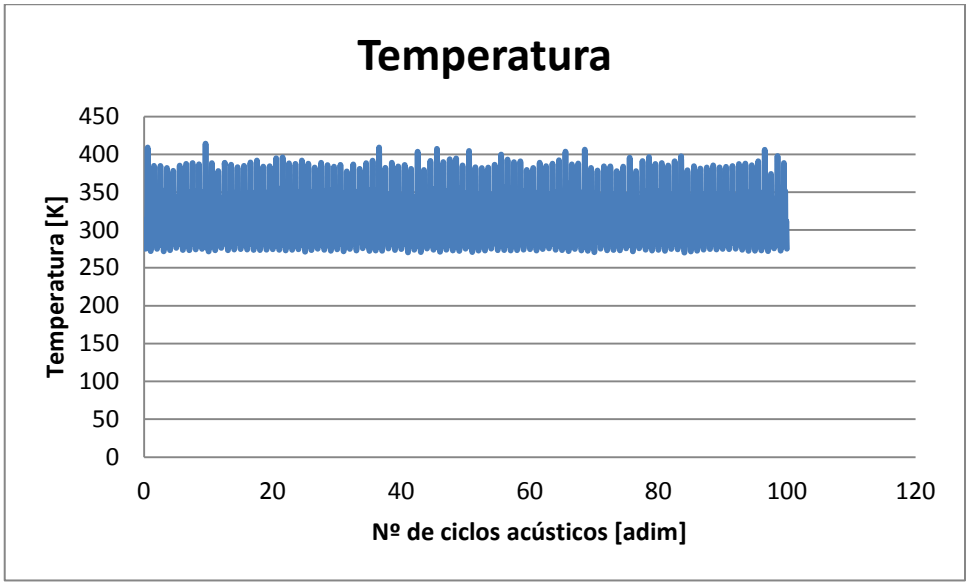
Amplitud de ultrasonido 0,4 bar:



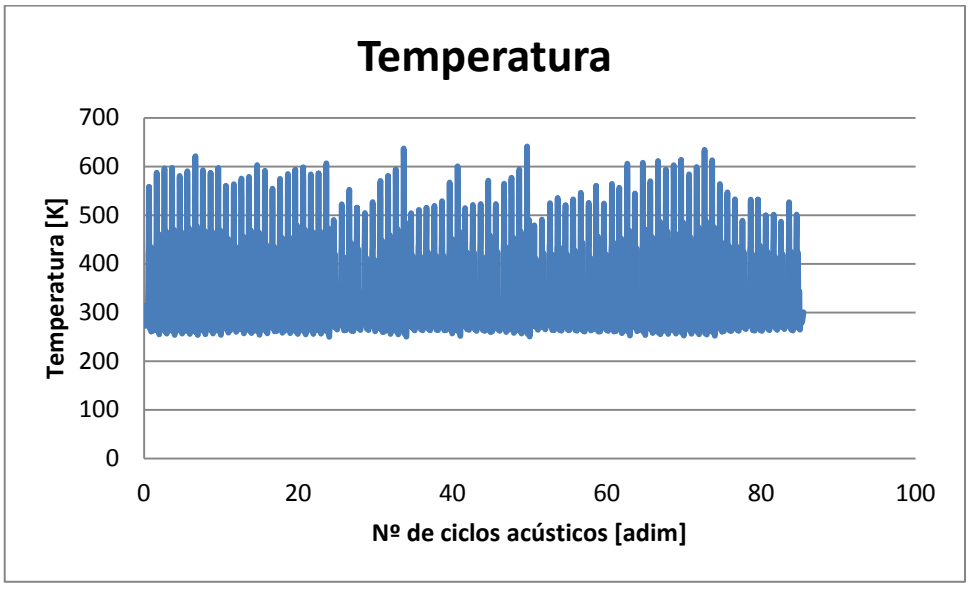
Amplitud de ultrasonido 0,5 bar:



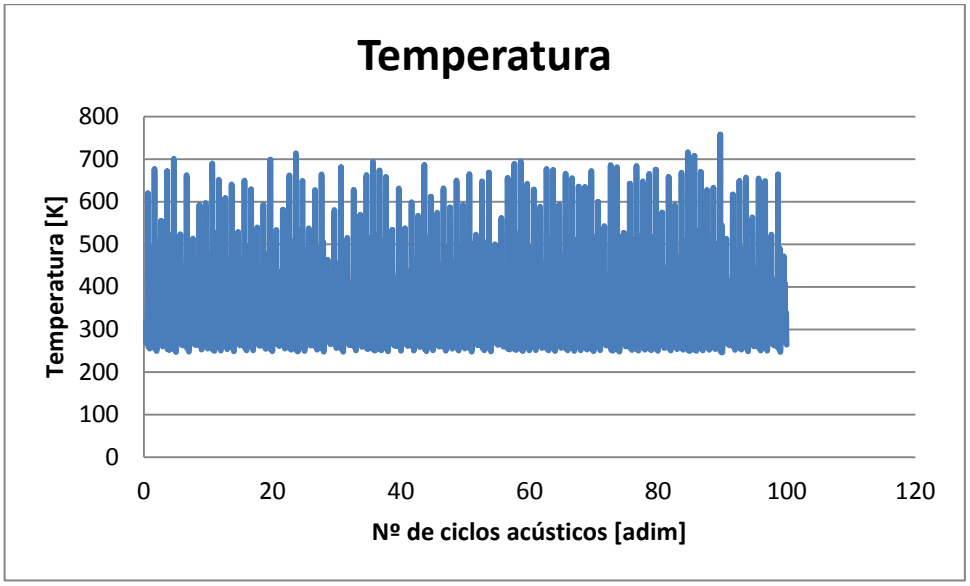
Amplitud de ultrasonido 0,6 bar:



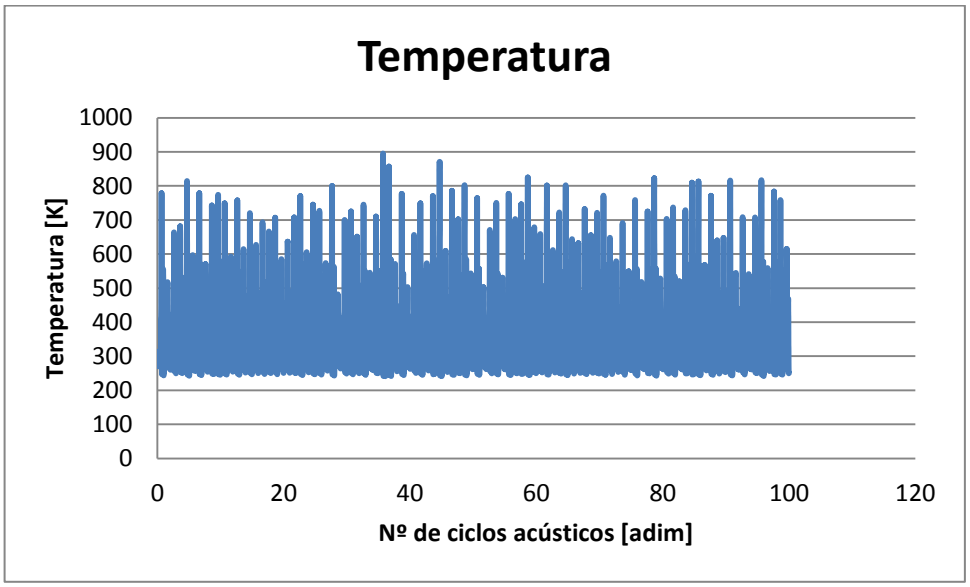
Amplitud de ultrasonido 0,75 bar:



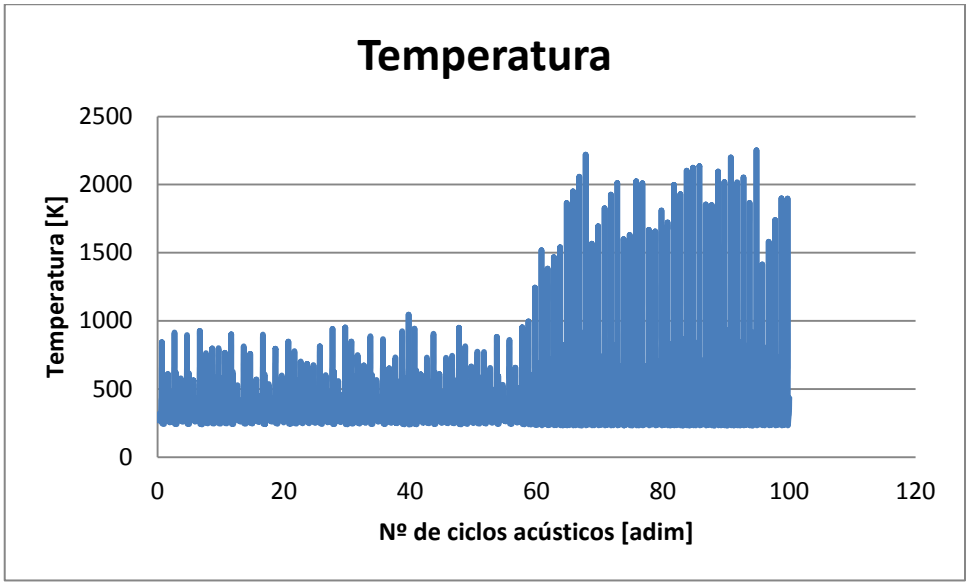
Amplitud de ultrasonido 0,8 bar:



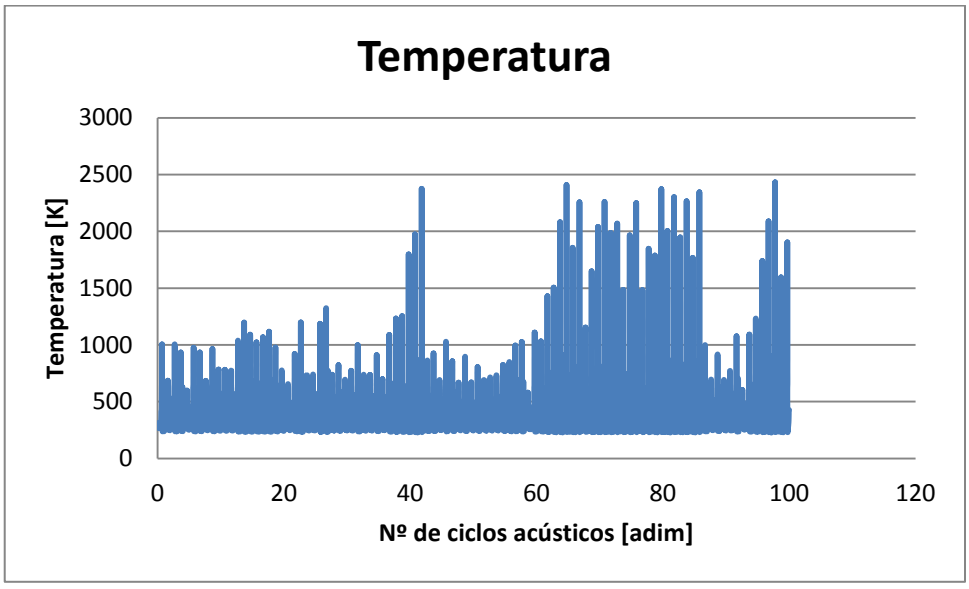
Amplitud de ultrasonido 0,85 bar:



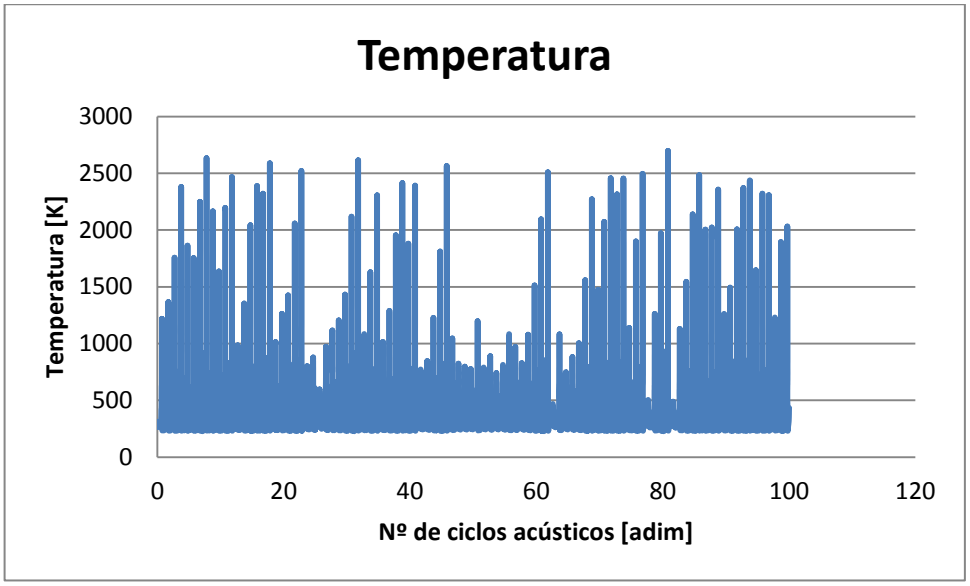
Amplitud de ultrasonido 0,9 bar:



Amplitud de ultrasonido 0,95 bar:



Amplitud de ultrasonido 1 bar:



ANEXO IX

En el siguiente anexo se adjuntan las tablas de datos de todos los estudios paramétricos expuestos en el apartado de resultados, con las siguientes tablas se han elaborado las gráficas de dicho apartado.

Temperatura inicial del líquido:

T	Cdest	Ccoque
273,15	0,176683	0,145222
283,15	0,177446	0,150863
288,15	0,17724	0,149919
293,15	0,176841	0,146723
298,15	0,175537	0,137291
303,15	0,173578	0,123664
308,15	0,169007	0,101977
313,15	0,158275	0,0761413
318,15	0,143162	0,0577016
323,15	0,115347	0,0385669
328,15	0,0775594	0,022472
333,15	0,0429229	0,0117688
336,15	0,0309263	0,00854558
338,15	0,125368	0,0418757
340,15	0,175702	0,142644
341,15	0,174404	0,13109
343,15	0,0555043	0,01455
346,15	0,166881	0,0944677
348,15	0,145577	0,0625056
349,15	0,120637	0,043804
350,15	0,101357	0,0339451
353,15	0,0297257	0,008834
358,15	0,00222625	0,00136344
363,15	0,00101815	0,00100646
368,15	0,00100004	0,00100002
373,15	0,001	0,001
450,15	0,001	0,001

Frecuencia de ultrasonido:

Frecuencia	Cdest	Ccoque
100	0,0457501	0,0126537
115	0,00451148	0,00190597
120	0,00300566	0,00152519
125	0,174418	0,140348
128	0,179152	0,174909

130	0,18024	0,180644
132	0,178732	0,168527
135	0,17871	0,165774
140	0,175118	0,135727
150	0,158275	0,0761413
175	0,0253586	0,00700498
200	0,173835	0,149469
225	0,203252	0,246628
226	0,202995	0,246425
227	0,202481	0,246105
228	0,201884	0,245666
229	0,20127	0,245176
230	0,200745	0,244745
232	0,199779	0,243755
240	0,195806	0,238441
250	0,190702	0,226681
260	0,186125	0,209062
270	0,181859	0,184485
275	0,179355	0,166407
300	0,162157	0,0834653
325	0,0884602	0,0266824
350	0,0210023	0,00597355
375	0,00379726	0,00172116
400	0,00137929	0,00110532
450	0,00101161	0,00100367
500	0,00100044	0,00100015

Presión inicial del líquido:

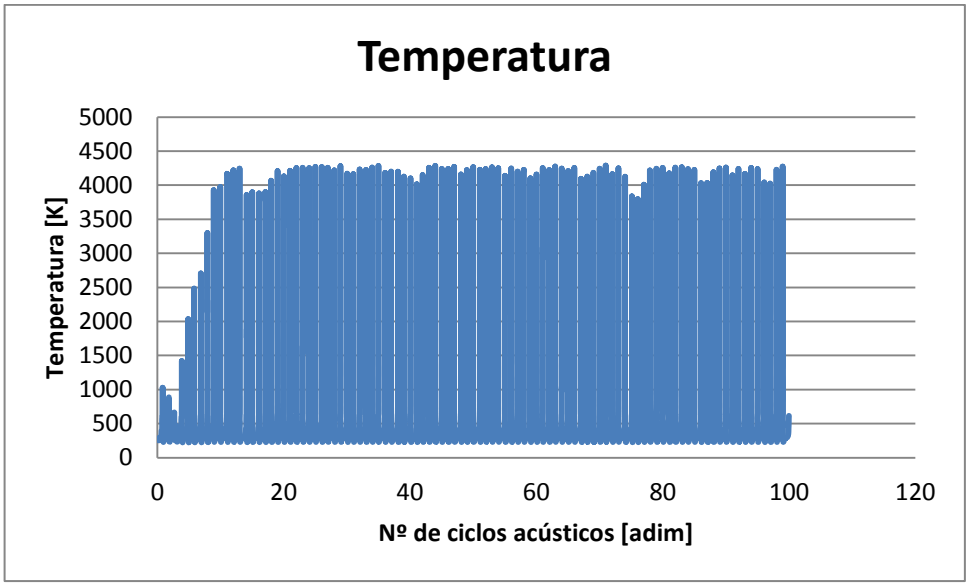
Presión	Cdest	Ccoque
1	0,158275	0,0761413
1,05	0,142947	0,0575481
1,1	0,0644452	0,0174163
1,15	0,00100171	0,00100058
1,2	0,00100016	0,00100006
1,25	0,00100001	0,00100001
1,3	0,001	0,001
1,5	0,001	0,001
2	0,001	0,001
2,5	0,001	0,001
3	0,001	0,001

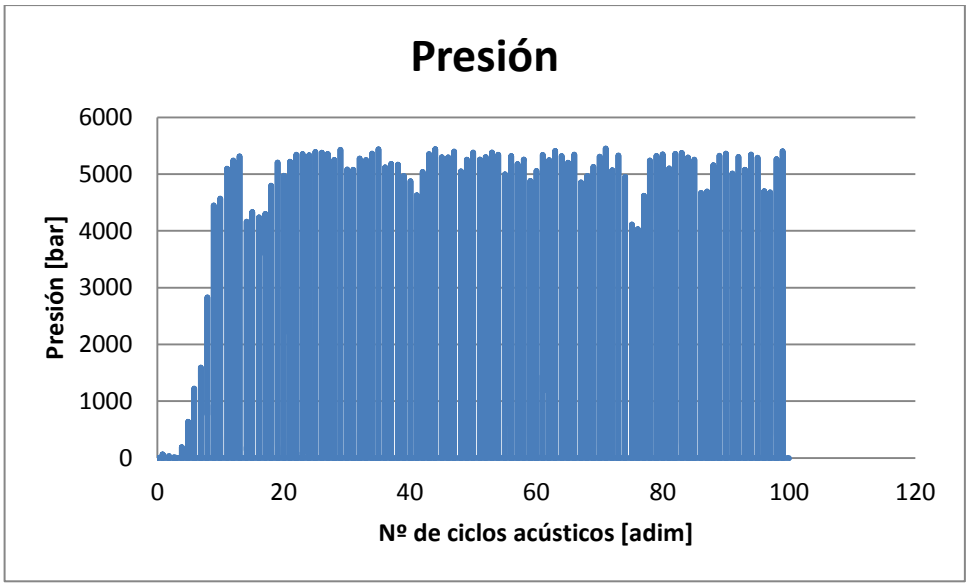
Amplitud de ultrasonido:

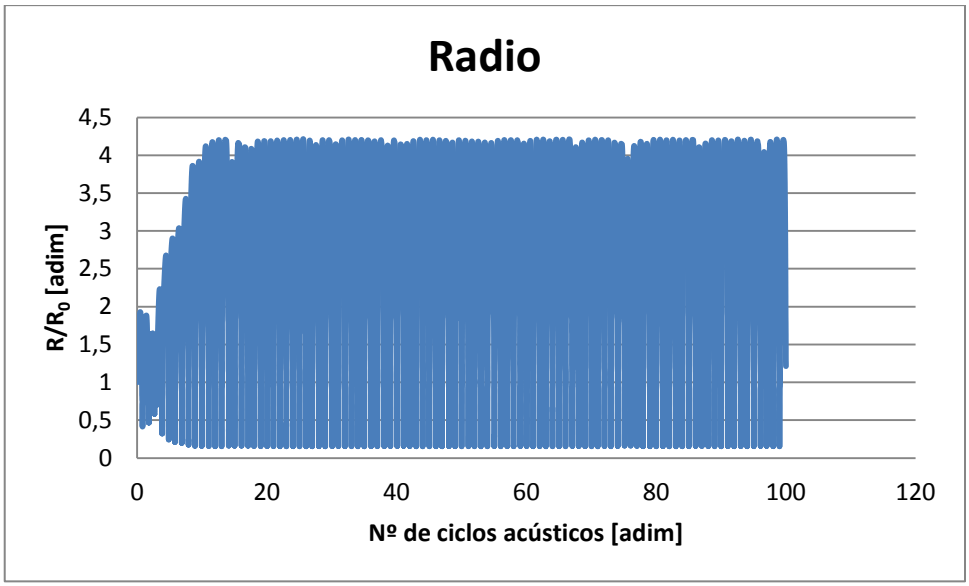
Amplitud	Cdest	Ccoque
0,01	0,001	0,001
0,1	0,001	0,001
0,15	0,001	0,001
0,25	0,001	0,001
0,4	0,001	0,001
0,5	0,001	0,001
0,6	0,001	0,001
0,75	0,001	0,001
0,8	0,001	0,001
0,85	0,00100009	0,00100004
0,9	0,0165461	0,00458967
0,95	0,123189	0,043072
1	0,158275	0,0761413

Anexo X

A continuación se encuentran las gráficas de la evolución del radio de la burbuja, los picos de presión, los picos de temperatura y las concentraciones de los productos finales, para la combinación de los valores óptimos de las variables de los análisis paramétricos.






Radio
4,5
4
3,5
3
2,5
2
1,5
1
0,5
0
R/R0 [adim]
0
20
40
60
80
100
120
Nº de ciclos acústicos [adim]

ANEXO XI

En el siguiente anexo se adjuntan las tablas de datos de todos los estudios paramétricos para la elaboración de la gráfica del estudio concentración versus temperatura de la constante cinética media.

Temperatura inicial del líquido:

T	Kdes	Tmed
273,15	270,55	1306,50016
283,15	289,925	1312,13327
288,15	286,597	1311,1896
293,15	276,694	1308,3237
298,15	248,99	1299,80028
303,15	211,686	1286,90188
308,15	161,555	1265,98366
313,15	110,561	1237,74849
318,15	79,5448	1214,23968
323,15	50,5346	1183,27305
328,15	28,3965	1146,13568
333,15	14,5189	1105,7464
336,15	10,4767	1087,112
338,15	57,7952	1192,27089
340,15	266,086	1305,15236
341,15	231,663	1294,03653
343,15	19,3574	1122,71006
346,15	158,825	1264,68731
348,15	122,457	1245,23145
349,15	74,5744	1209,73688
350,15	60,1139	1194,93329
353,15	18,5053	1120,02071
358,15	0,44314	934,457923
363,15	0,00655615	787,205912
368,15	1,04E-05	634,30161
373,15	3,33E-09	510,488457
450,15	4,29E-18	341,141268

Frecuencia de ultrasonido:

Frecuencia	Kdest media	T media Kdest
100	10,0856	1084,9801
115	1,05431	971,854321

120	0,66239	951,436601
125	216,607	1288,71253
128	323,218	1321,08546
130	349,92	1327,70001
132	310,255	1317,70023
135	306,856	1316,79215
140	227,866	1292,72314
150	110,561	1237,74849
175	9,16064	1079,62695
200	412,537	1341,6313
225	2109,93	1497,40213
226	2090,75	1496,42999
227	2060,9	1494,90172
228	2016,38	1492,58669
229	1971,89	1490,22898
230	1938,58	1488,43366
232	1866,48	1484,45512
240	1594,29	1468,13467
250	1272,28	1445,38912
260	992,162	1421,12147
270	758,158	1395,77244
275	629,289	1378,7397
300	249,048	1299,81897
325	73,7093	1208,92616
350	15,8859	1110,99798
375	2,88043	1019,14758
400	0,538235	942,593223
450	0,0254143	829,224597
500	0,00119167	740,017141

Presión inicial del líquido:

Presión	Kmedia	Tmedia
1	110,561	1237,74849
1,05	78,6907	1213,48398
1,1	20,7707	1126,94589
1,15	0,00126504	741,574872
1,2	0,00012213	685,144671
1,25	7,45E-06	627,968163
1,3	2,20E-10	478,928538
1,5	8,84E-14	406,599773
2	2,15E-19	325,361033
2,5	5,96E-20	319,030764
3	6,00E-20	319,060686

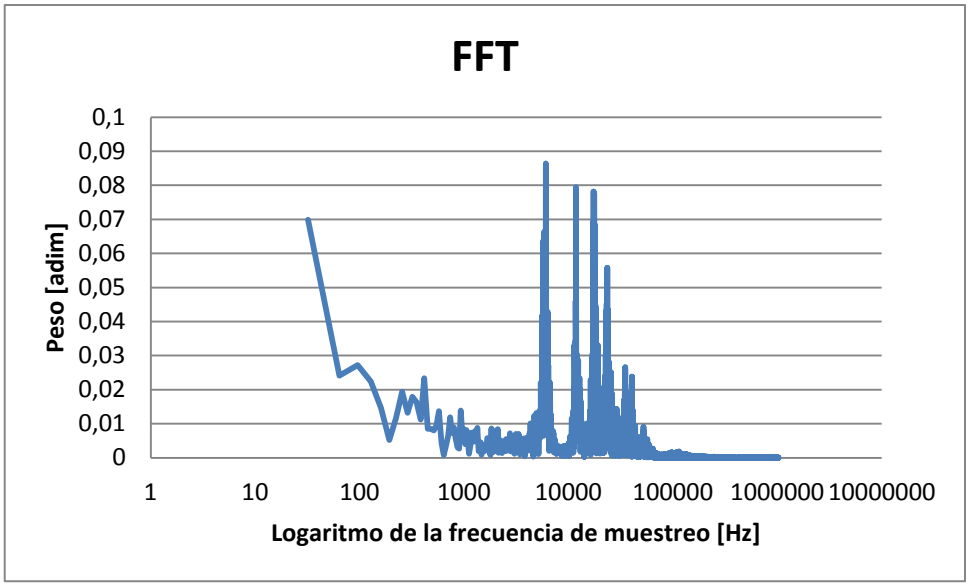
Amplitud de ultrasonido:

Amplitud	Kmedia	Tmedia
0,01	5,88E-22	298,156363
0,1	6,96E-22	298,870226
0,15	8,45E-22	299,695708
0,25	1,51E-21	302,198266
0,4	2,00E-20	313,834434
0,5	4,76E-19	329,399813
0,6	6,77E-11	466,432926
0,75	3,45E-09	510,941107
0,8	1,87E-08	532,817361
0,85	6,61E-05	671,723077
0,9	4,44E+00	1040,9209
0,95	5,63E+01	1190,55191
1	110,561	1237,74849

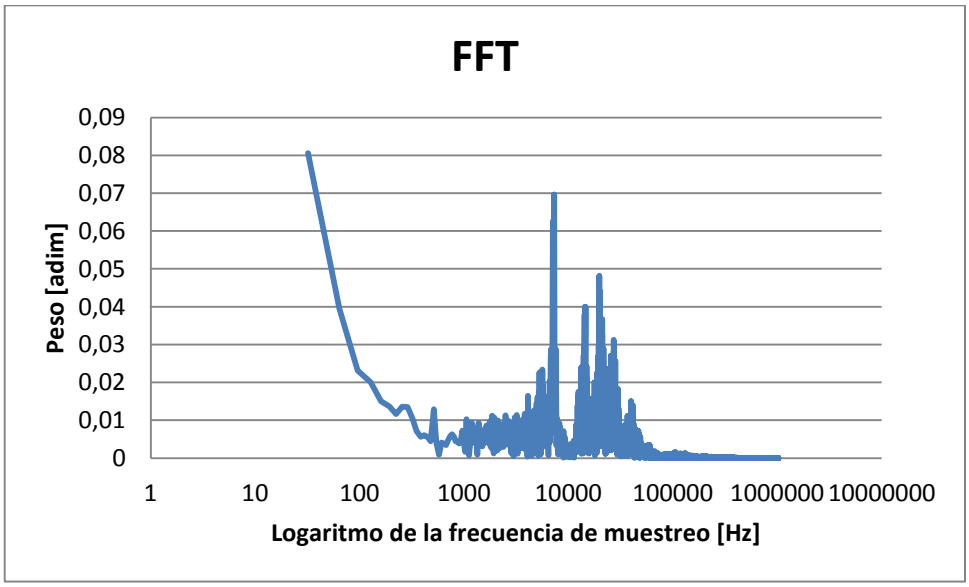
Anexo XII

En este anexo se muestran las gráficas resultantes del estudio de la transformada rápida de Fourier para la temperatura de burbuja del análisis paramétrico de la frecuencia de ultrasonido para las frecuencias que tienen concentraciones finales de productos intermedias y que, por tanto, gráficas de FFT intermedias entre los casos extremos reflejados en el análisis de los datos, gráficas nº9 y nº10.

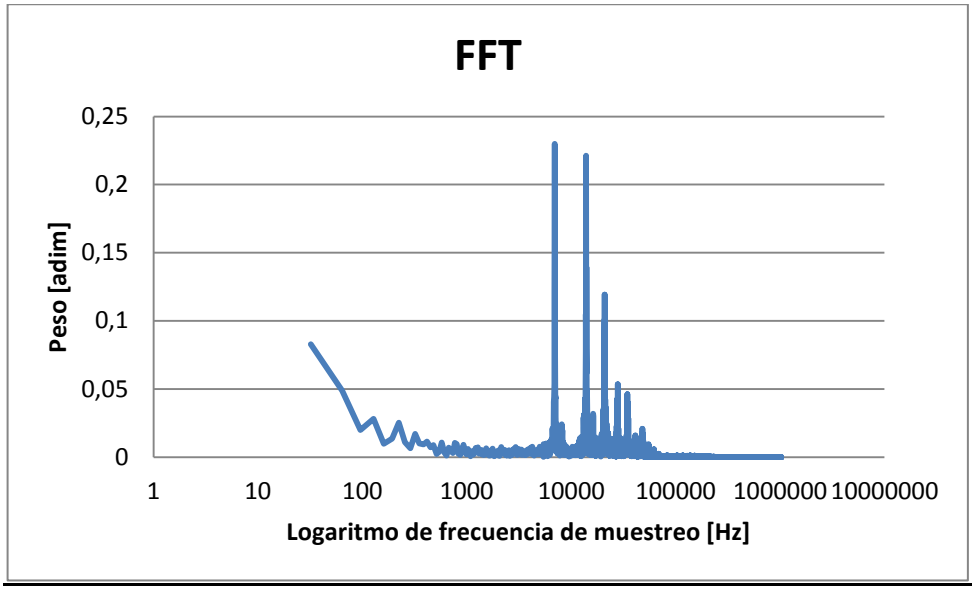
Frecuencia 100 kHz:



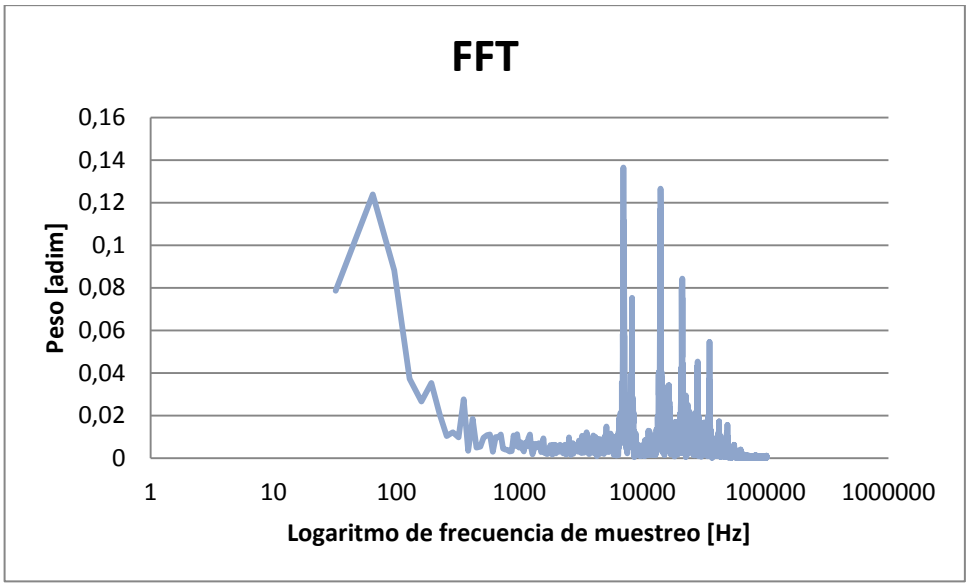
Frecuencia 115 kHz:



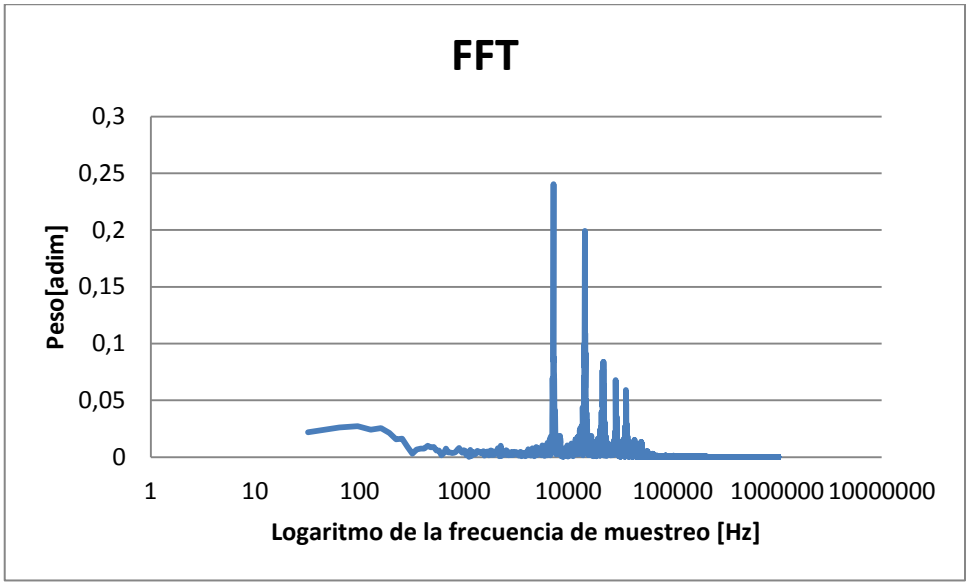
Frecuencia 125 kHz:



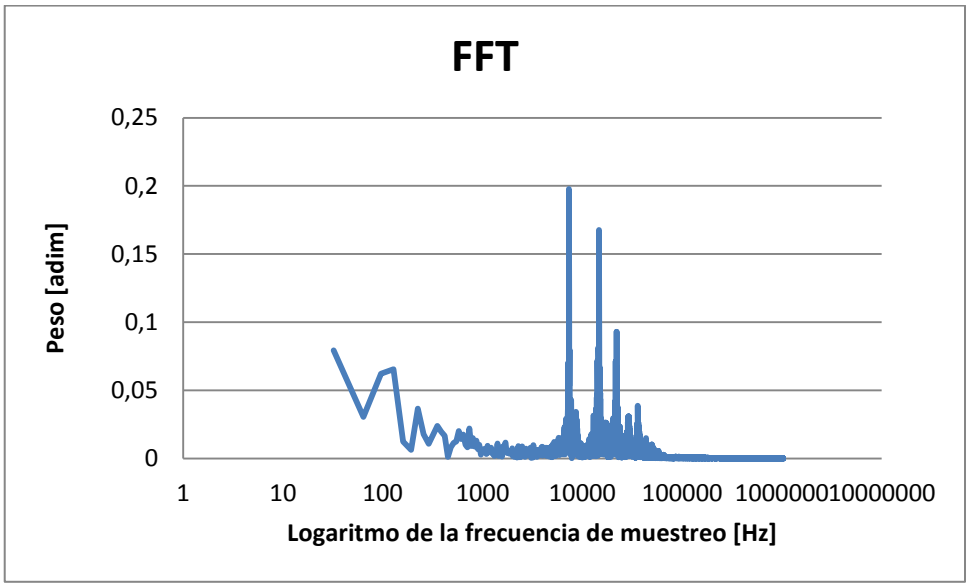
Frecuencia 128 kHz:



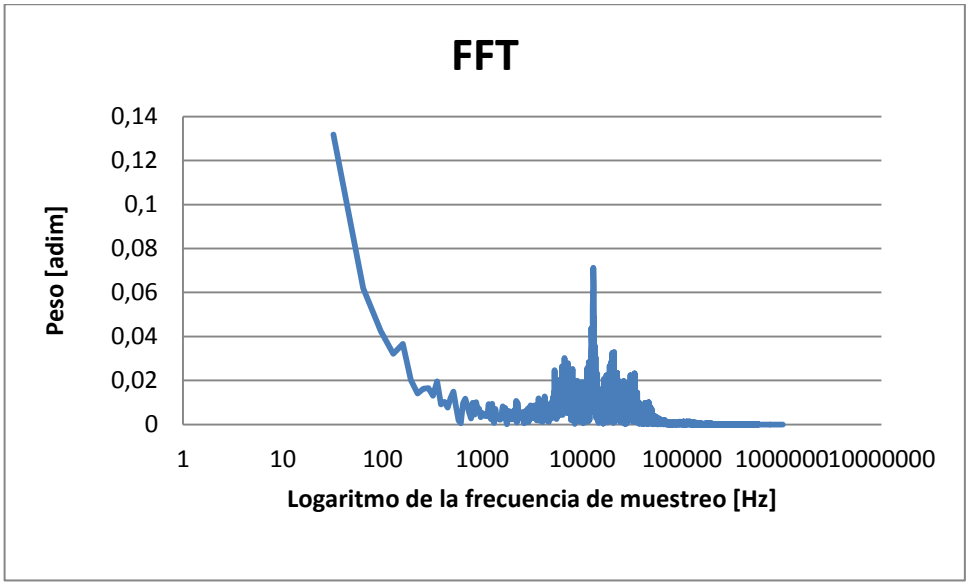
Frecuencia 132 kHz:



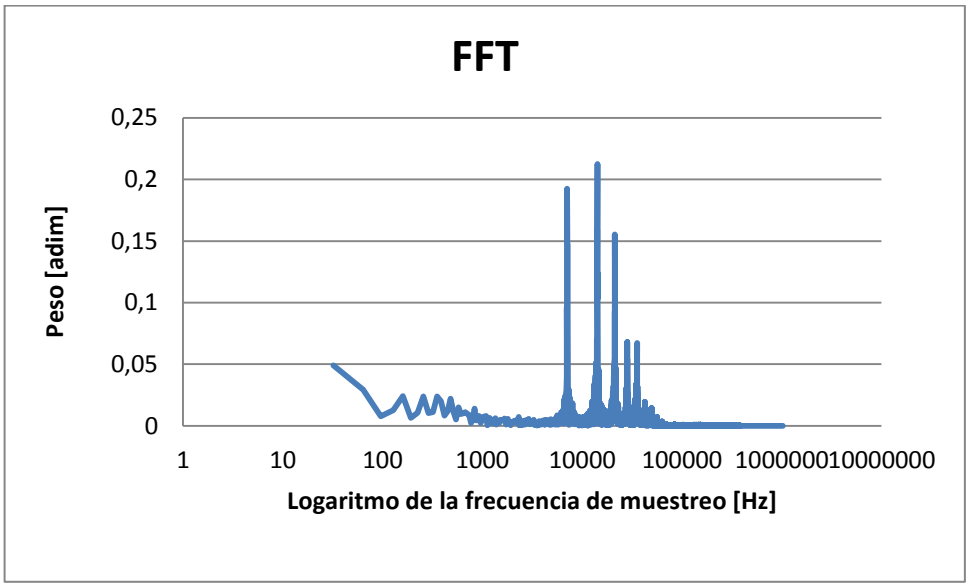
Frecuencia 135 kHz:



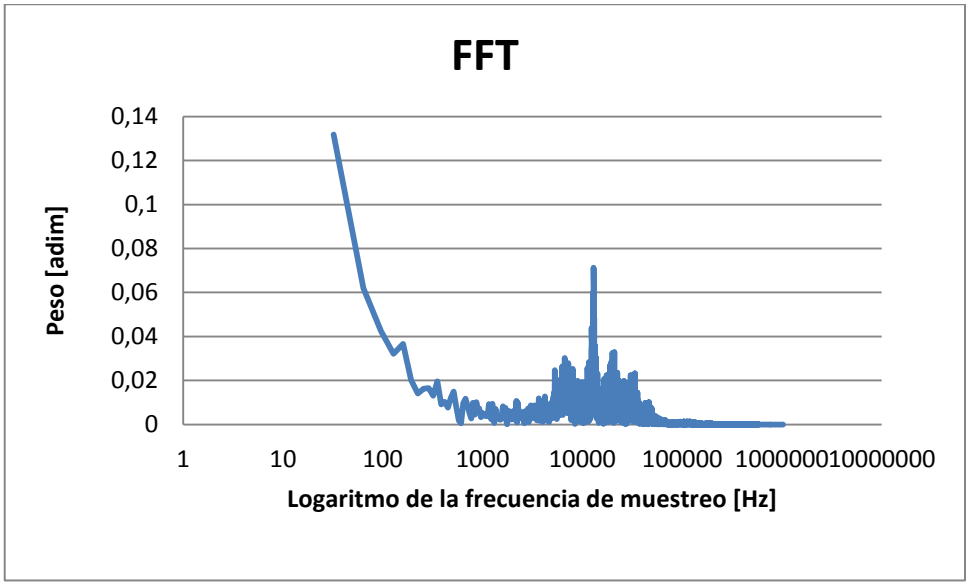
Frecuencia 140 kHz:



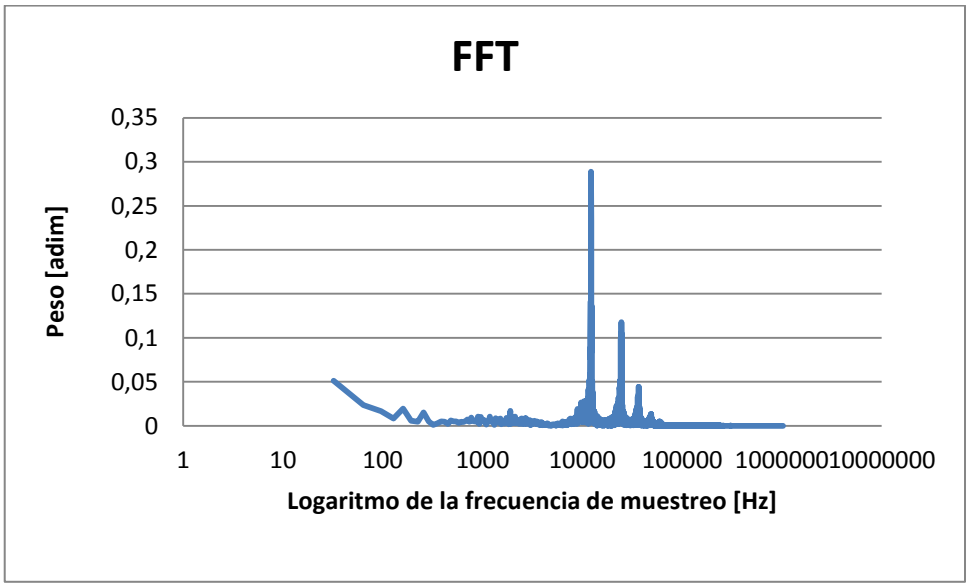
Frecuencia 150 kHz:



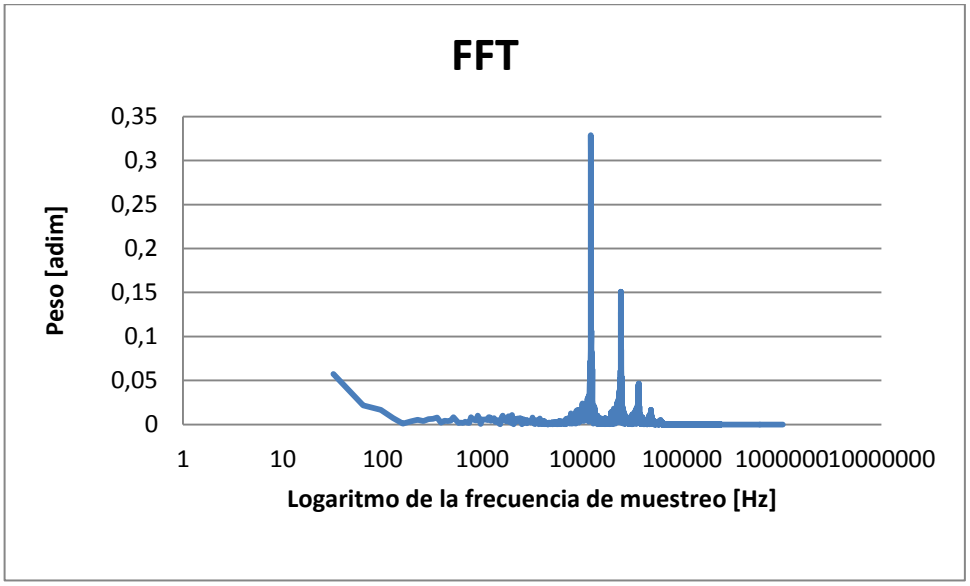
Frecuencia 200 kHz:



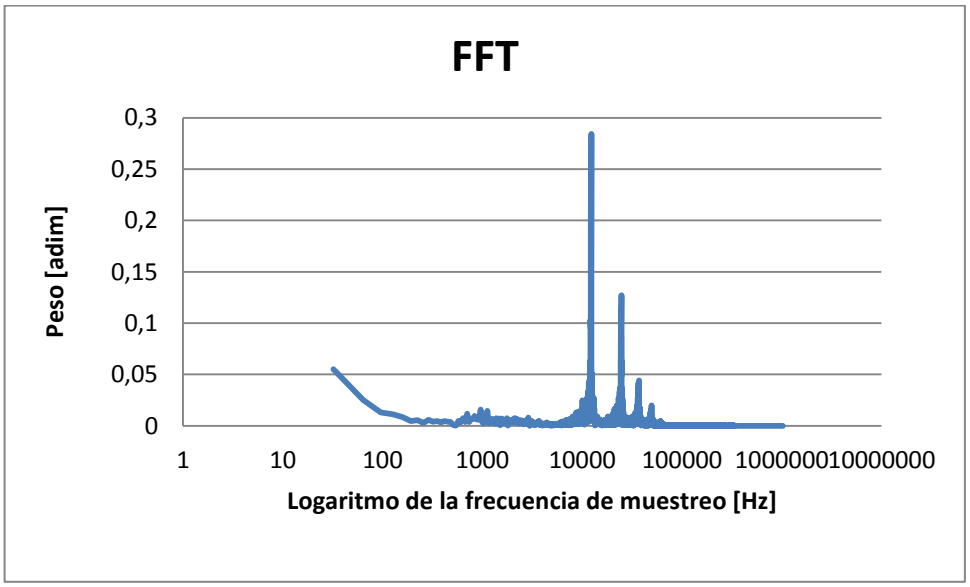
Frecuencia 227 kHz:



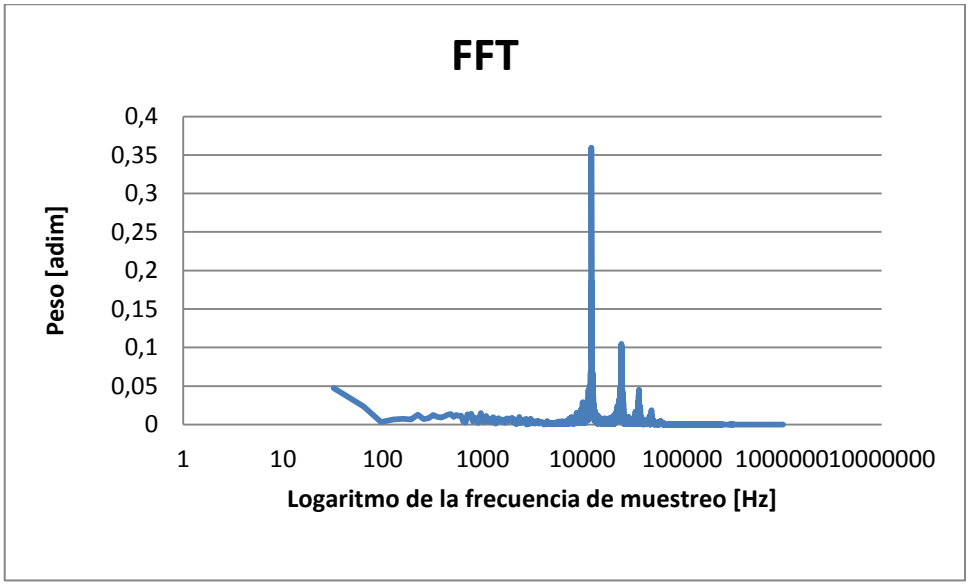
Frecuencia 228 kHz:



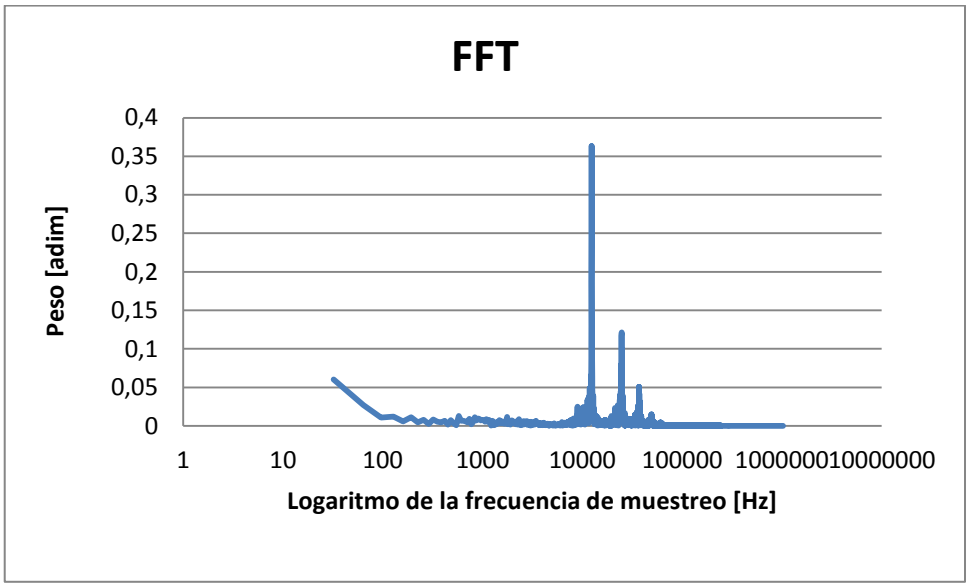
Frecuencia 229 kHz:



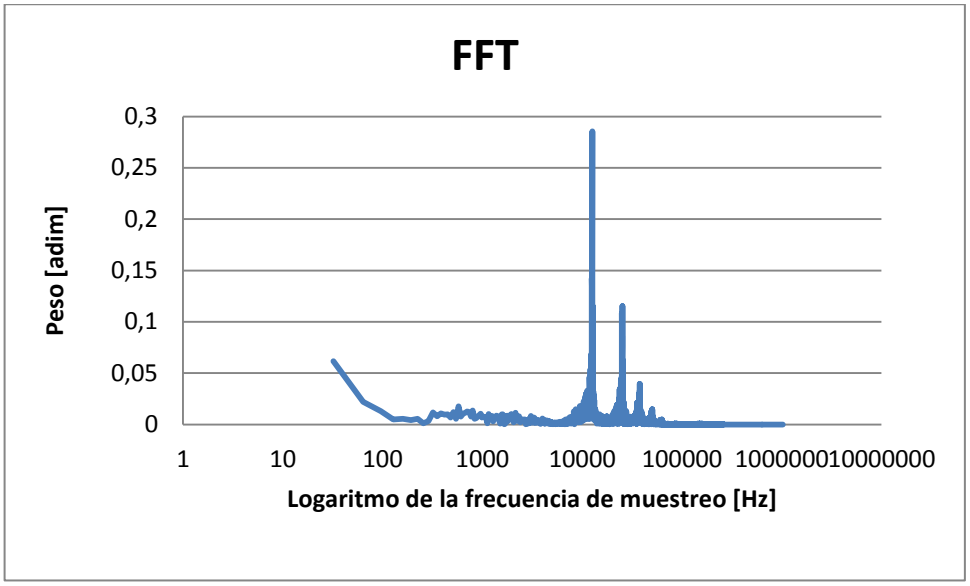
Frecuencia 230 kHz:



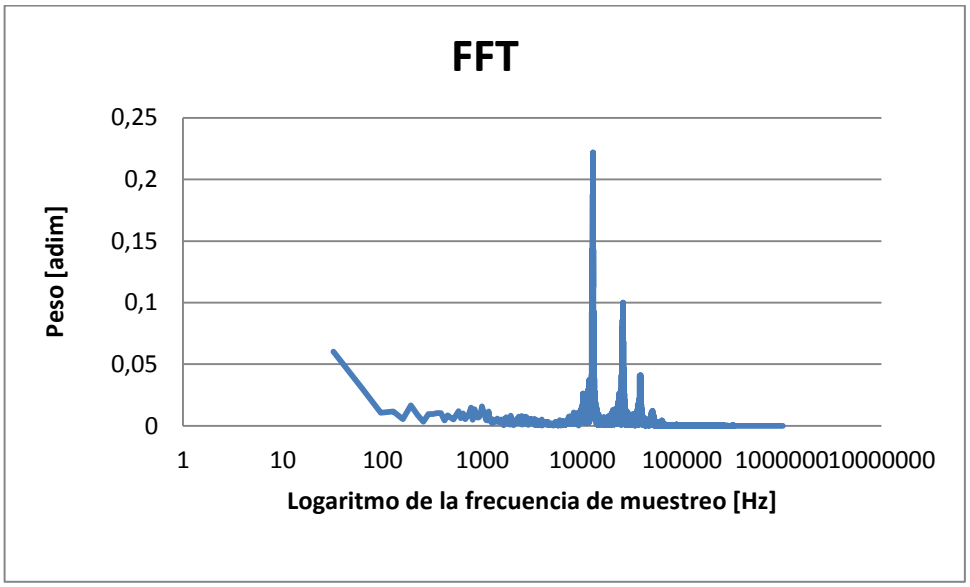
Frecuencia 232 kHz:



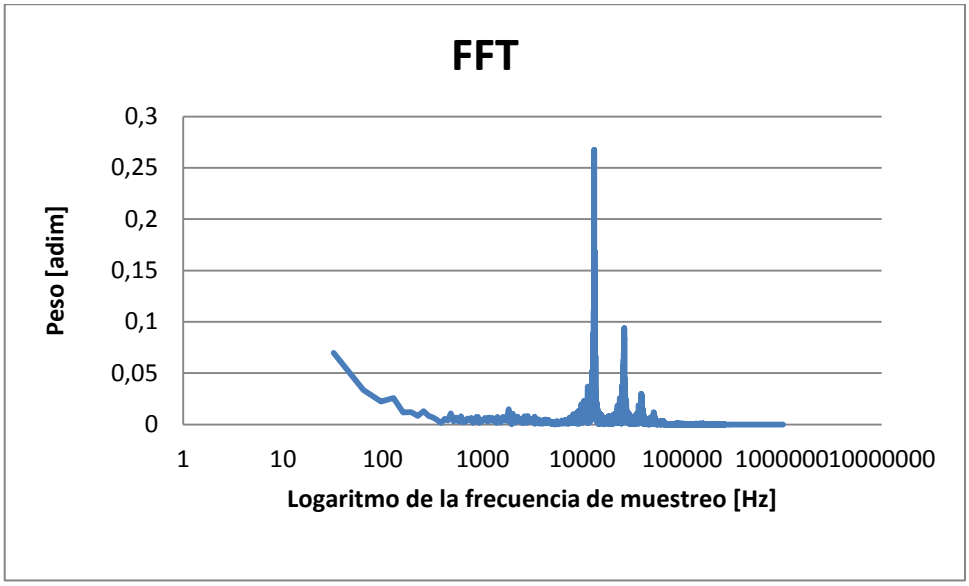
Frecuencia 240 kHz:



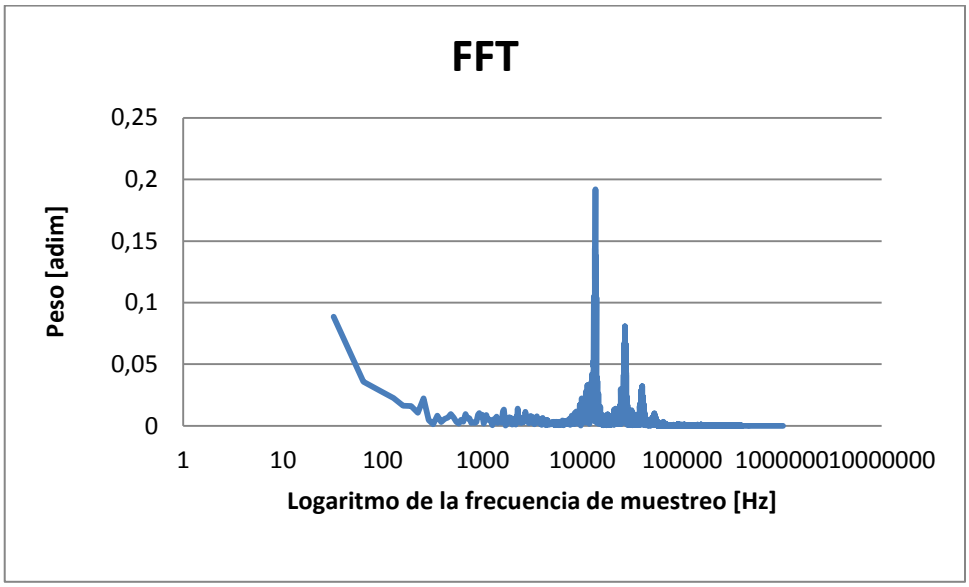
Frecuencia 250 kHz:



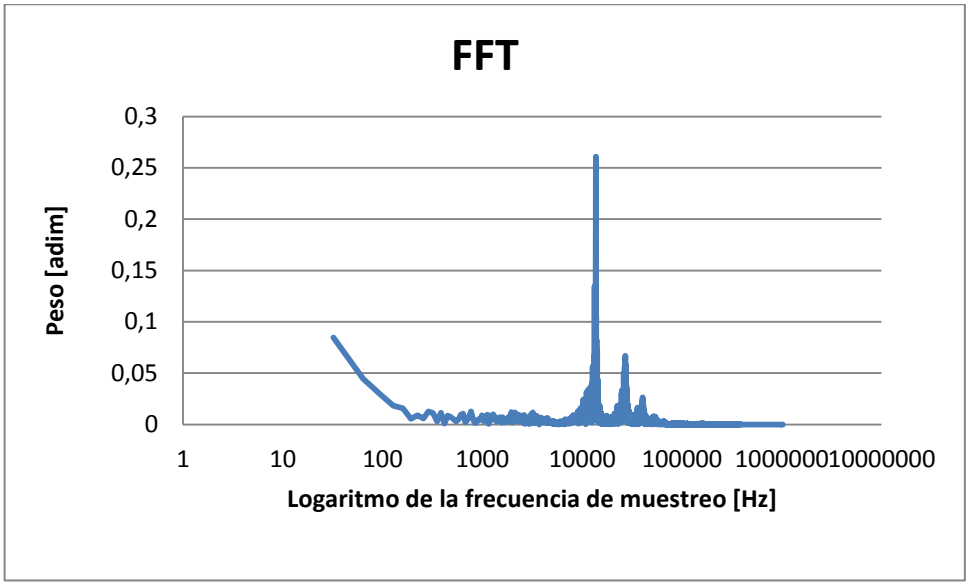
Frecuencia 260 kHz:



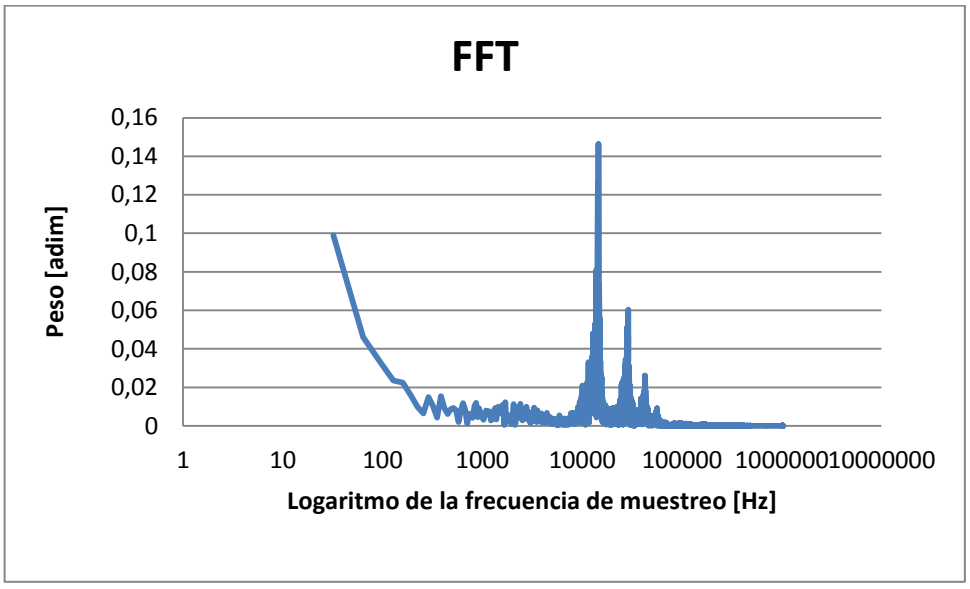
Frecuencia 270 kHz:



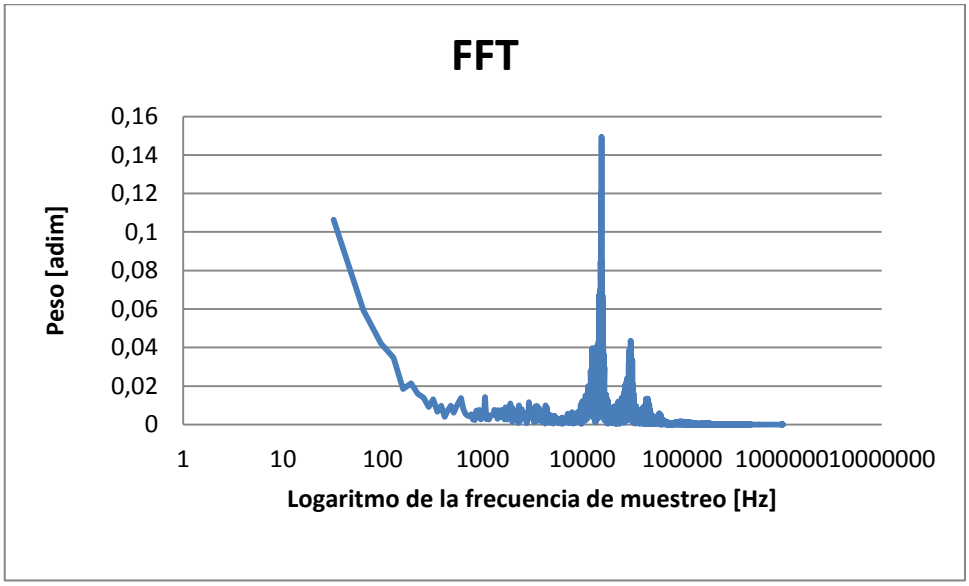
Frecuencia 275 kHz:



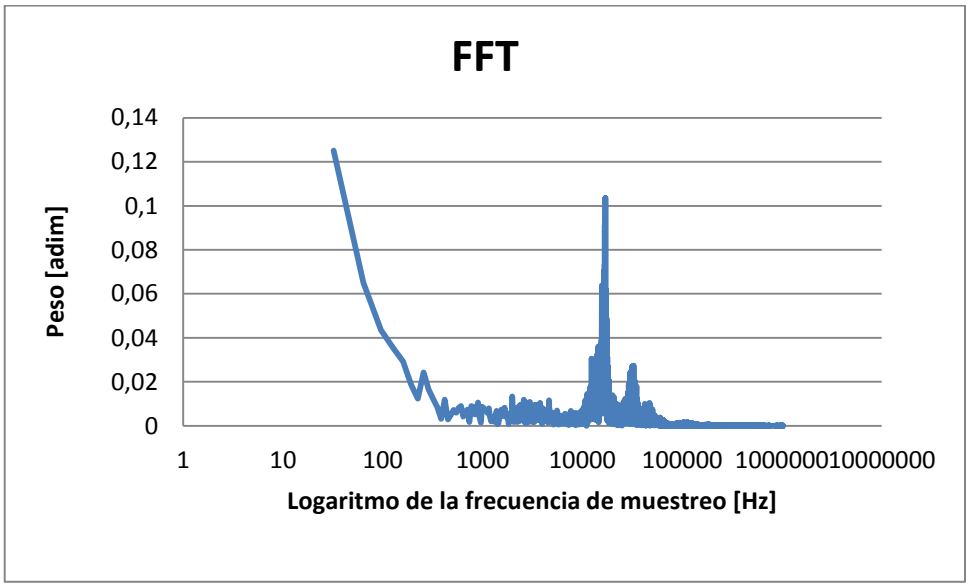
Frecuencia 300 kHz:



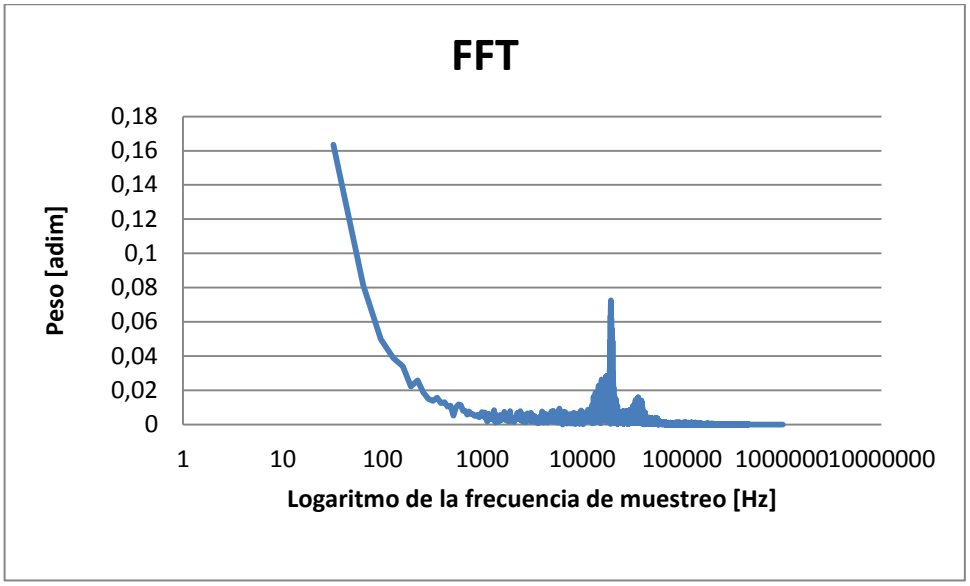
Frecuencia 325 kHz:



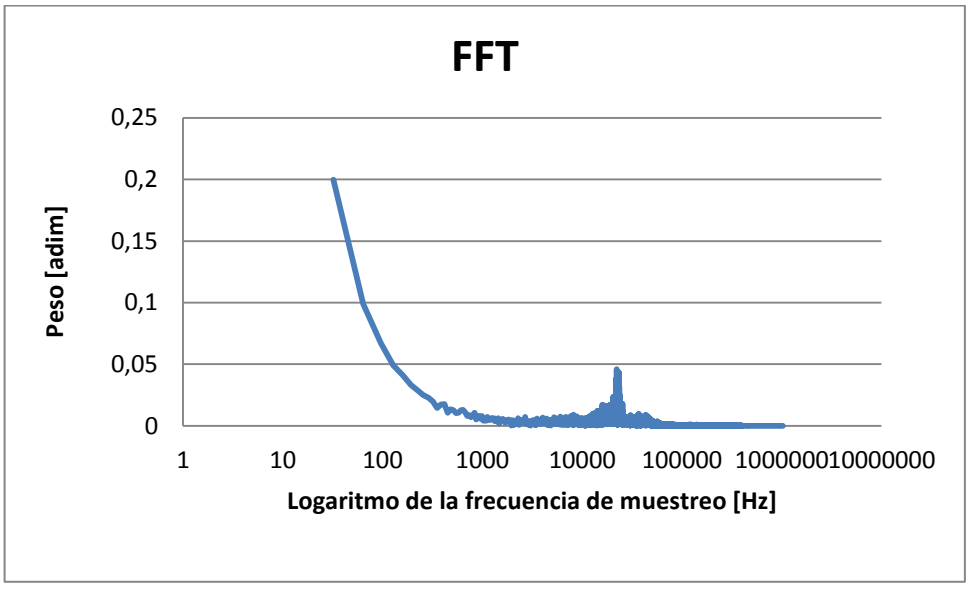
Frecuencia 350 kHz:



Frecuencia 400 kHz:



Frecuencia 450 kHz:



Frecuencia 500 kHz: