GRP2

Dietmar G. Schrausser

2023

Overview

Coordinate system and graphics tool for SCHRAUSSER-MAT (Schrausser, 2022). Application for MS Windows (Schrausser, 2023).

C++ Source of main functions

```
//------------------------------------------------------------------------| GRP2Dlg.cpp
//
//                                                                        | GRP2, Graphik
Dialog 2            (deutsch)  //
//                                                                        | von Dietmar
Schrausser, (C) SCHRAUSSER 2011  //
//

#include "stdafx.h"
#include "GRP2.hpp"
#include "GRP2Dlg.hpp"
#include "GRP2info.hpp"
#include "GRP2param1.hpp"
#include "GRP2param2.hpp"
#include "GRP2param3.hpp"
#include "GRP2param4.hpp"
#include "GRP2param5.hpp"
#include "GRP2diagrm1.hpp"
#include "GRP2diagrm2.hpp"
#include "GRP2diagrm3.hpp"
#include "GRP2diagrm4.hpp"
#include "GRP2diagrm5.hpp"
#include "GRP2diagrm6.hpp"
#include "GRP2fn1.hpp"
#include "GRP2fn2.hpp"
#include "GRP2fn3.hpp"
#include "GRP2fn4.hpp"
#include "GRP2prg.hpp"
#include "GRP2koord.hpp"
#include "GRP2splash.hpp"
#include "D:\_EIGENEDATEIEN_\1_LAUFENDES\1_SYSTEM\3_C_PROGRAMME\__H_C++_\DATACONV.HPP"
#include "D:\_EIGENEDATEIEN_\1_LAUFENDES\1_SYSTEM\3_C_PROGRAMME\__H_C_\FN_PQ.H"
#include "D:\_EIGENEDATEIEN_\1_LAUFENDES\1_SYSTEM\3_C_PROGRAMME\__H_C_\FN_SORT.H"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

RECT coord;

BOOL sw_x,    sw_y;    //Achsen_Beschriftung_FunktionsWerte
BOOL sw_x_0,  sw_y_0;
BOOL sw_xSw,  sw_ySw;  //Achsen_Beschriftung_SkalenWerte
BOOL sw_xSw_0,sw_ySw_0;
BOOL sw_xm,   sw_ym;   //Achsen_Beschriftung_MinMax
BOOL sw_xm_0, sw_ym_0;
BOOL sw_xA,   sw_yA;   //Achsen_
BOOL sw_xA_0, sw_yA_0;
BOOL sw_xV,   sw_yV;   //Achsen_Vektoren
BOOL sw_xV_0, sw_yV_0;
BOOL sw_xS,   sw_yS;   //Achsen_Skala
BOOL sw_xS_0, sw_yS_0;
BOOL sw_xK,   sw_yK;   //Achsen_Koordinaten
BOOL sw_xK_0, sw_yK_0;
BOOL sw_FK_0, sw_FK;   //funktionskurve
BOOL sw_xy_0, sw_xy;   //xy punkte
BOOL sw_xb,   sw_yb;   //achsen bezeichnungsschalter
BOOL sw_xb_0, sw_yb_0;
BOOL sw_emf;           //emf schalter
BOOL sw_emf_in;
BOOL sw_inv;           //inversfunktion f-1(x)
BOOL sw_mod_;          //funktionsmodusschalter
BOOL modus_;           //startmodusschalter
BOOL sw_csr;           //cursorschalter
BOOL sw_v0;            //vektorursprungspositionsschalter
BOOL sw_mkoord_A, sw_mkoord_V;        //manueller Koordinateneingabe schalter
BOOL sw_splash;
BOOL sw_xbz, sw_ybz; //manueller achsenbezeichnungsschalter
BOOL sw_Fxy; //Funktionsmatrix Fxy Ansicht schalter
BOOL sw_Theta; //Theta Ansicht schalter
BOOL sw_Log; //Log Ansicht schalter
BOOL sw_status; //Statusleisten Ansichts schalter
int nlog=1; //logindex

BOOL sw_integral;//integralschalter
int int_n;//ordnung n

BOOL sw_differential;//differentialschalter
int diff_n;//ordnung n

BOOL sw_kgl;//mittelungsschalter
int kgl_i;//iterationen i

BOOL sw_fxy=0;//funktionsmatrix neu setzenschalter

int file_ind=1;//dateineusetzungsindex

BOOL sw_drw=1; //funktions-renderschalter
BOOL dynrnd; //Dynamisch Rendern
BOOL filestr; //Filestream Rendern

BOOL wnd_pos, log_, csr_, csr_0; //Programmeinstellungsschalter

short sw_sz=0; //size marker
short sw_bd=0; //mouseschaltermarker
short p_e_sw, p_e_sw_0; //p modus schalter
short p_p_sw, p_p_sw_0; //

short sw_pq, sw_pq_0;
short sw_Grdx, sw_Grdx_0; //Grid schalter
short sw_Grdy, sw_Grdy_0; //

// Funktionsvektoren (x),(y) von Funktionsmatrix (F)
double FVx_[33010], FVy_[33010];     //nicht in GRP2Dlg.hpp definieren, felddimensionierung
heikel!!

int scrx, scry;

int posVx, posVy, posVx_0, posVy_0; //vektorkoordinatenausgabe positions shift
int posAXx, posAXy, posAXx_0, posAXy_0; //Achsenskalenwert positions shift
int posAYx, posAYy, posAYx_0, posAYy_0; //

int posBXx, posBXy, posBXx_0, posBXy_0; //Achsenbezeichnungs positions shift
int posBYx, posBYy, posBYx_0, posBYy_0; //

int posFXx, posFXy, posFXx_0, posFXy_0; //Funktionswert positions shift
int posFYx, posFYy, posFYx_0, posFYy_0; //

int posSCx, posSCx_0; //Skalenmarkierungs positions shift
int posSCy, posSCy_0; //

int posX, posX_0; //Diagramm x-shift
int posY, posY_0; //Diagramm y-shift

float frmX, frmX_0; //Diagramm x-form shift
float frmY, frmY_0; //Diagramm y-form shift

float n_;
float mv1y, mv1x;
float mv2y, mv2x;
float corx;
float corx0;
float cory;
float cory0;
float sc0x;
float sc0y;
float sc, sc_0, scy, scy_0; // skalierungs variablen
float tlg_x, tlg_x_0, tlg_y, tlg_y_0;//int skalendelta
float Vp_x, Ap_x, Vp_y, Ap_y;

float mVx, mAx, mVy, mAy; //manuelle koordinatenbestimmungs variablen
float mVx_0, mAx_0, mVy_0, mAy_0; //manuelle koordinatenbestimmungs variablen
float Ax_m,Ay_m;//,Vx_m,Vy_m

 //Konfidenzintervalle
float CI_Pp, CI_Pp_0, ci_zp, ci_tp; //Fp
float CI_Pr, CI_Pr_0, ci_zr, ci_tr; //rxy
float CI_Pe, CI_Pe_0, ci_ze, ci_te, ci_ze_0; //Fe

BOOL Dx_,Dy_;

BOOL ds_x,   ds_y,   ds_xSw,   ds_ySw,   ds_xk,   ds_yk;  //dezimalstellen
BOOL ds_x_0, ds_y_0, ds_xSw_0, ds_ySw_0, ds_xk_0, ds_yk_0;//

BOOL tab_;     //parametertabellenposition
BOOL tab_diag; //diagrammtabellenposition
BOOL tab_fn;   //funktionstabellenposition

BOOL linB_Ax,   linB_Ay,   linB_Vx,   linB_Vy,   linB_Fn,   linB_FnP;  //linienbreite
BOOL linB_Ax_0, linB_Ay_0, linB_Vx_0, linB_Vy_0, linB_Fn_0, linB_FnP_0;//

BOOL linB_Grdx,   linB_Grdy;  //grid linienbreite
BOOL linB_Grdx_0, linB_Grdy_0;//

BOOL linB_r,   linB_ri,   linB_sr,   linB_sri,   linB_sR,   linB_sRi,   linB_sxy,   linB_syx,
linB_sgxy,   linB_sgyx;   //rxy linienbreite
BOOL linB_r_0, linB_ri_0, linB_sr_0, linB_sri_0, linB_sR_0, linB_sRi_0, linB_sxy_0,
linB_syx_0,linB_sgxy_0, linB_sgyx_0; //
BOOL linB_am,   linB_sd,   linB_sgam,   linB_sdg,   linB_a3,   linB_ag3,   linB_sga3,
linB_a4,   linB_ag4,   linB_sga4;   //Fp linienbreite
BOOL linB_am_0, linB_sd_0, linB_sgam_0, linB_sdg_0, linB_a3_0, linB_ag3_0, linB_sga3_0,
linB_a4_0, linB_ag4_0, linB_sga4_0; //
BOOL linB_e,   linB_x,   linB_x0,   linB_x1,   linB_s0,   linB_s1;   //Fe linienbreite
BOOL linB_e_0, linB_x_0, linB_x0_0, linB_x1_0, linB_s0_0, linB_s1_0; //

BOOL r_q, ri_q, syx_q, sxy_q, sgyx_q ,sgxy_q, sr_q, sri_q, sR_q,sRi_q;
//Theta darstellungsschalter
BOOL r_q_0, ri_q_0, syx_q_0, sxy_q_0, sgyx_q_0,sgxy_q_0, sr_q_0, sri_q_0, sR_q_0,sRi_q_0;  //
BOOL am_q, sd_q, sdg_q, sgam_q, a3_q,ag3_q, sa3g_q, a4_q, ag4_q,sa4g_q;                    //
BOOL am_q_0, sd_q_0, sdg_q_0, sgam_q_0, a3_q_0,ag3_q_0, sa3g_q_0, a4_q_0, ag4_q_0,sa4g_q_0;//
BOOL e_q,   x_q,   x0_q,   x1_q,   s0_q,  s1_q;                    //
BOOL e_q_0, x_q_0, x0_q_0, x1_q_0, s0_q_0,s1_q_0;//

BOOL mod_Ax,   mod_Ay;  //Achsenmodus
BOOL mod_Ax_0, mod_Ay_0;//
BOOL mod_Vx,   mod_Vy;  //vektormodus
BOOL mod_Vx_0, mod_Vy_0;//
BOOL mod_Fn,   mod_Fn_0; //Funktionskurvenmodus
BOOL mod_FnP,  mod_FnP_0;//

BOOL mod_Grdx,   mod_Grdy;  //grid modus
BOOL mod_Grdx_0, mod_Grdy_0;//

BOOL mod_r,   mod_ri,   mod_sr,   mod_sri,   mod_sR,   mod_sRi,   mod_sxy,   mod_syx,
mod_sgxy,   mod_sgyx;   //rxy modus
BOOL mod_r_0, mod_ri_0, mod_sr_0, mod_sri_0, mod_sR_0, mod_sRi_0, mod_sxy_0,
mod_syx_0,mod_sgxy_0, mod_sgyx_0; //
BOOL mod_am,   mod_sd,   mod_sgam,   mod_sdg,   mod_a3,   mod_ag3,   mod_sga3,   mod_a4,
mod_ag4,   mod_sga4;   //Fp modus
BOOL mod_am_0, mod_sd_0, mod_sgam_0, mod_sdg_0, mod_a3_0, mod_ag3_0, mod_sga3_0, mod_a4_0,
mod_ag4_0, mod_sga4_0; //
BOOL mod_e,   mod_x,   mod_x0,   mod_s0,   mod_x1,   mod_s1;   //Fe modus
BOOL mod_e_0, mod_x_0, mod_x0_0, mod_s0_0, mod_x1_0, mod_s1_0; //

//int dt_xy; //xy punktgrösse

BOOL fb_hg, fb_hg_0; //hintergrundfarbe
BOOL fb_K, fb_K_0;   //Kurvenfarbe
BOOL fb_P, fb_P_0;   //Punktfarbe
BOOL fb_Ax, fb_Ay, fb_Ax_0, fb_Ay_0; //Achsenfarbe
BOOL fb_Vx, fb_Vy, fb_Vx_0, fb_Vy_0; //Vektorfarbe

BOOL fb_Grdx,   fb_Grdy;  //grid farbe
BOOL fb_Grdx_0, fb_Grdy_0;//

BOOL fb_r,   fb_ri,   fb_sr,   fb_sri,   fb_sR,   fb_sRi,   fb_sxy,   fb_syx,  fb_sgxy,
fb_sgyx;   //rxy farben
BOOL fb_r_0, fb_ri_0, fb_sr_0, fb_sri_0, fb_sR_0, fb_sRi_0, fb_sxy_0, fb_syx_0,fb_sgxy_0,
fb_sgyx_0; //
BOOL fb_am,   fb_sd,   fb_sgam,   fb_sdg,   fb_a3,   fb_ag3,   fb_sga3,   fb_a4,   fb_ag4,
fb_sga4;   //Fp farben
BOOL fb_am_0, fb_sd_0, fb_sgam_0, fb_sdg_0, fb_a3_0, fb_ag3_0, fb_sga3_0, fb_a4_0, fb_ag4_0,
fb_sga4_0; //
BOOL fb_e,   fb_s0,   fb_s1,   fb_x0,   fb_x1,   fb_x;   //Fe farben
BOOL fb_e_0, fb_s0_0, fb_s1_0, fb_x0_0, fb_x1_0, fb_x_0; //

//schrift
//funktion
CString fn_x_fon,   fn_y_fon;                    //art
CString fn_x_fon_0, fn_y_fon_0;
BOOL    fn_x_fb,    fn_y_fb;                     //farbe
BOOL    fn_x_fb_0,  fn_y_fb_0;
BOOL    fn_x_H,     fn_x_W,   fn_y_H,   fn_y_W;  //grösse
BOOL    fn_x_H_0,   fn_x_W_0, fn_y_H_0, fn_y_W_0;
//achsen
CString Ax_fon,     Ay_fon;                      //art
CString Ax_fon_0,   Ay_fon_0;
BOOL    Ax_fb,      Ay_fb;                       //farbe
BOOL    Ax_fb_0,    Ay_fb_0;
BOOL    Ax_H,       Ax_W,   Ay_H,   Ay_W;        //grösse
BOOL    Ax_H_0,     Ax_W_0, Ay_H_0, Ay_W_0;
//vektoren
CString V_fon;            //art
CString V_fon_0;
BOOL    V_fb;             //farbe
BOOL    V_fb_0;
BOOL    V_H,        V_W;  //grösse
BOOL    V_H_0,      V_W_0;

BOOL rxy_D, rxy_D_0;           //Regressionsüberlappungsschalter
BOOL rxy_, ryx_, rxy_0, ryx_0; //Regressionslinienschalter
BOOL sxy_, syx_, sxy_0, syx_0; //Vorhersagefehlerlinienschalter
BOOL s1xy_, s1yx_, s1xy_0, s1yx_0; //Geschätzter Vorhersagefehlerlinienschalter
BOOL srxy_, sryx_, srxy_0, sryx_0; //Geschätzter Korrelationsfehlerlinienschalter
BOOL srx_, sry_, srx_0, sry_0;     //Geschätzter Regressionsfehlerlinienschalter

BOOL a3_, a3_0; //a3 linienschalter
BOOL a4_, a4_0;   //a4 linienschalter
BOOL ag3_, ag3_0;   //a'3 linienschalter
BOOL ag4_, ag4_0;    //a'4 linienschalter
BOOL am_ , am_0;   //am linienschalter
BOOL sd_ , sd_0;   //sd linienschalter
BOOL sdg_ , sdg_0;    //sd' linienschalter
BOOL sga3_, sga3_0;   //s'a3 linienschalter
BOOL sga4_ , sga4_0; //s'a4 linienschalter
BOOL sgam_, sgam_0; //s'am linienschalter

BOOL e_, e_0; //e linienschalter
BOOL xc_, xc_0; //x linienschalter
BOOL x0_, x0_0; //x linienschalter
BOOL x1_, x1_0; //x linienschalter
BOOL s0_, s0_0; //x linienschalter
BOOL s1_, s1_0; //x linienschalter

CString fil;
CString xBz, yBz, xBz_0, yBz_0; //manuelle achsenbezeichnung

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
       //{{AFX_DATA_INIT(CAboutDlg)
       //}}AFX_DATA_INIT
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
       CDialog::DoDataExchange(pDX);
```

```
        //{{AFX_DATA_MAP(CAboutDlg)
        //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
        //{{AFX_MSG_MAP(CAboutDlg)
        //}}AFX_MSG_MAP
END_MESSAGE_MAP()


CGRP2Dlg::CGRP2Dlg(CWnd* pParent )
        : CDialog(CGRP2Dlg::IDD, pParent)
{
        //{{AFX_DATA_INIT(CGRP2Dlg)
        //}}AFX_DATA_INIT
        m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
    m_Csr1  = AfxGetApp()->LoadCursor(IDC_CURSOR1);
        m_Csr2  = AfxGetApp()->LoadCursor(IDC_CURSOR2);
        m_Csr3  = AfxGetApp()->LoadCursor(IDC_CURSOR3);
}

void CGRP2Dlg::DoDataExchange(CDataExchange* pDX)
{
        CDialog::DoDataExchange(pDX);
        //{{AFX_DATA_MAP(CGRP2Dlg)
        //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CGRP2Dlg, CDialog)
        //{{AFX_MSG_MAP(CGRP2Dlg)
        ON_WM_SYSCOMMAND()
        ON_WM_PAINT()
        ON_WM_QUERYDRAGICON()
        ON_WM_SIZE()
        ON_WM_MOUSEMOVE()
        ON_WM_TIMER()
        ON_COMMAND(ID_FUNKTIONSMATRIX_FFNEN, OnFunktionsmatrixFfnen)
        ON_COMMAND(ID_FUNKTION_EMFSPEICHERN, OnFunktionEmfspeichern)
        ON_WM_MOUSEWHEEL()
        ON_WM_CLOSE()
        ON_COMMAND(ID_MODUS_FP, OnModusFp)
        ON_COMMAND(ID_MODUS_FX, OnModusFx)
        ON_COMMAND(ID_MODUS_RXY, OnModusRxy)
        ON_COMMAND(ID_FUNKTION_EMFFFNEN, OnFunktionEmfffnen)
        ON_COMMAND(ID_MODUS_INV, OnModusInv)
        ON_COMMAND(ID_EINSTELLUNGEN_Parameter, OnEINSTELLUNGENParameter)
        ON_COMMAND(ID_INFO, OnInfo)
        ON_COMMAND(ID_EINSTELLUNGEN_CURSOR_SCHWARZ, OnEinstellungenCursorSchwarz)
        ON_COMMAND(ID_EINSTELLUNGEN_CURSOR_WEISS, OnEinstellungenCursorWeiss)
        ON_COMMAND(ID_EINSTELLUNGEN_Diagramm, OnEINSTELLUNGENDiagramm)
        ON_COMMAND(ID_EINSTELLUNGEN_Funktion, OnEINSTELLUNGENFunktion)
        ON_COMMAND(ID_EINSTELLUNGEN_KOORDINATEN, OnEinstellungenKoordinaten)
        ON_COMMAND(ID_EINSTELLUNGEN_APP, OnEinstellungenApp)
        ON_WM_RBUTTONUP()
        ON_WM_LBUTTONUP()
        ON_WM_LBUTTONDOWN()
        ON_WM_RBUTTONDOWN()
        ON_COMMAND(ID_MODUS_EPSILON, OnModusEpsilon)
        ON_COMMAND(ID_EINSTELLUNGEN_CURSOR_ACHSENVERSCHUB, OnEinstellungenCursorAchsenverschub)
        ON_COMMAND(ID_EINSTELLUNGEN_CURSOR_VEKTORENVERSCHUB,
OnEinstellungenCursorVektorenverschub)
        ON_COMMAND(ID_FUNKTION_SPEICHERNUNTER, OnFunktionSpeichernunter)
        ON_COMMAND(ID_ANSICHT_FUNKTIONSMATRIXFXY, OnAnsichtFunktionsmatrixfxy)
        ON_COMMAND(ID_MODUS_FZX, OnModusFzx)
        ON_COMMAND(ID_ANSICHT_THETAFENSTERQ, OnAnsichtThetafensterq)
        ON_COMMAND(ID_FUNKTION_SPEICHERN, OnFunktionSpeichern)
        ON_COMMAND(ID_FUNKTION_EMFSPEICHERNUNTER, OnFunktionEmfspeichernunter)
        ON_COMMAND(ID_ANSICHT_GRP2LOG, OnAnsichtGrp2log)
        ON_COMMAND(ID_ANSICHT_STATUSLEISTE, OnAnsichtStatusleiste)
        //}}AFX_MSG_MAP
END_MESSAGE_MAP()

void CGRP2Dlg::cmdline(CString f){fil=f;}; // cmdline arg

BOOL CGRP2Dlg::OnInitDialog()
{
        CDialog::OnInitDialog();
```

```
     CWinApp* pApp = AfxGetApp(); // ini profil lesen

sw_x  =       pApp->GetProfileInt("Achsen","x_Beschriftung_Werte",0);
sw_y  =       pApp->GetProfileInt("Achsen","y_Beschriftung_Werte",0);
sw_xm =       pApp->GetProfileInt("Achsen","x_Beschriftung_MinMax",0);
sw_ym =       pApp->GetProfileInt("Achsen","y_Beschriftung_MinMax",0);
sw_xA =       pApp->GetProfileInt("Achsen","x_",1);
sw_yA =       pApp->GetProfileInt("Achsen","y_",1);
sw_xb=        pApp->GetProfileInt("Achsen","X_Bezeichnung",1);
sw_yb=        pApp->GetProfileInt("Achsen","Y_Bezeichnung",1);
sw_xV =       pApp->GetProfileInt("Achsen","x_Vektor",1);
sw_yV =       pApp->GetProfileInt("Achsen","y_Vektor",1);
sw_xS =       pApp->GetProfileInt("Achsen","x_Skala",1);
sw_yS =       pApp->GetProfileInt("Achsen","y_Skala",1);
sw_Grdx =     pApp->GetProfileInt("Achsen","x_Grid",0);
sw_Grdy =     pApp->GetProfileInt("Achsen","y_Grid",0);
sw_xSw=       pApp->GetProfileInt("Achsen","x_Skalenwerte",1);
sw_ySw=       pApp->GetProfileInt("Achsen","y_Skalenwerte",1);
sw_xK =       pApp->GetProfileInt("Achsen","x_Koordinate",1);
sw_yK =       pApp->GetProfileInt("Achsen","y_Koordinate",1);
ds_x=         pApp->GetProfileInt("Achsen","x_Werte_Dezimalstellen",3);
ds_y=         pApp->GetProfileInt("Achsen","y_Werte_Dezimalstellen",3);
ds_xSw=       pApp->GetProfileInt("Achsen","x_Skalenwerte_Dezimalstellen",2);
ds_ySw=       pApp->GetProfileInt("Achsen","y_Skalenwerte_Dezimalstellen",2);
ds_xk=        pApp->GetProfileInt("Achsen","x_Koordinaten_Dezimalstellen",2);
ds_yk=        pApp->GetProfileInt("Achsen","y_Koordinaten_Dezimalstellen",2);
fb_Ax=        pApp->GetProfileInt("Achsen","X_Farbe",8421504);
fb_Ay=        pApp->GetProfileInt("Achsen","Y_Farbe",8421504);
fb_Vx=        pApp->GetProfileInt("Achsen","X_Vektor_Farbe",8421504);
fb_Vy=        pApp->GetProfileInt("Achsen","Y_Vektor_Farbe",8421504);
posVx=        pApp->GetProfileInt("Achsen","X_Vektor_Koordinaten_Position",3);
posVy=        pApp->GetProfileInt("Achsen","Y_Vektor_Koordinaten_Position",-11);
posAXx=       pApp->GetProfileInt("Achsen","X_Skalenwerte_Position_x",0);
posAXy=       pApp->GetProfileInt("Achsen","X_Skalenwerte_Position_y",3);
posAYx=       pApp->GetProfileInt("Achsen","Y_Skalenwerte_Position_x",-26);
posAYy=       pApp->GetProfileInt("Achsen","Y_Skalenwerte_Position_y",-8);
posBXx=       pApp->GetProfileInt("Achsen","X_Bezeichnungs_Position_x",0);
posBXy=       pApp->GetProfileInt("Achsen","X_Bezeichnungs_Position_y",0);
posBYx=       pApp->GetProfileInt("Achsen","Y_Bezeichnungs_Position_x",0);
posBYy=       pApp->GetProfileInt("Achsen","Y_Bezeichnungs_Position_y",0);
posFXx=       pApp->GetProfileInt("Achsen","X_Beschriftung_Werte_Position_x",0);
posFXy=       pApp->GetProfileInt("Achsen","X_Beschriftung_Werte_Position_y",-12);
posFYx=       pApp->GetProfileInt("Achsen","Y_Beschriftung_Werte_Position_x",6);
posFYy=       pApp->GetProfileInt("Achsen","Y_Beschriftung_Werte_Position_y",-8);
posSCx=       pApp->GetProfileInt("Achsen","X_Skalierungs_Position",0);
posSCy=       pApp->GetProfileInt("Achsen","Y_Skalierungs_Position",0);
fb_r=         pApp->GetProfileInt("Funktion","rxy_Farbe",8421504);
fb_ri=        pApp->GetProfileInt("Funktion","ryx_Farbe",8421504);
fb_sr=        pApp->GetProfileInt("Funktion","srxy_Farbe",8421504);
fb_sri=       pApp->GetProfileInt("Funktion","sryx_Farbe",8421504);
fb_sR=        pApp->GetProfileInt("Funktion","sry_Farbe",8421504);
fb_sRi=       pApp->GetProfileInt("Funktion","srx_Farbe",8421504);
fb_sxy=       pApp->GetProfileInt("Funktion","sxy_Farbe",8421504);
fb_syx=       pApp->GetProfileInt("Funktion","syx_Farbe",8421504);
fb_sgxy=      pApp->GetProfileInt("Funktion","s'xy_Farbe",8421504);
fb_sgyx=      pApp->GetProfileInt("Funktion","s'yx_Farbe",8421504);
fb_am=        pApp->GetProfileInt("Funktion","am_Farbe",8421504);
fb_sd=        pApp->GetProfileInt("Funktion","sd_Farbe",8421504);
fb_sgam=      pApp->GetProfileInt("Funktion","s'am_Farbe",8421504);
fb_sdg=       pApp->GetProfileInt("Funktion","sd'_Farbe",8421504);
fb_a3=        pApp->GetProfileInt("Funktion","a3_Farbe",8421504);
fb_ag3=       pApp->GetProfileInt("Funktion","a3'_Farbe",8421504);
fb_sga3=      pApp->GetProfileInt("Funktion","s'a3_Farbe",8421504);
fb_a4=        pApp->GetProfileInt("Funktion","a4_Farbe",8421504);
fb_ag4=       pApp->GetProfileInt("Funktion","a4'_Farbe",8421504);
fb_sga4=      pApp->GetProfileInt("Funktion","s'a4_Farbe",8421504);
fb_e=         pApp->GetProfileInt("Funktion","e_Farbe",8421504);
fb_x=         pApp->GetProfileInt("Funktion","x_Farbe",8421504);
fb_x0=        pApp->GetProfileInt("Funktion","x0_Farbe",8421504);
fb_x1=        pApp->GetProfileInt("Funktion","x1_Farbe",8421504);
fb_s0=        pApp->GetProfileInt("Funktion","s0_Farbe",8421504);
fb_s1=        pApp->GetProfileInt("Funktion","s1_Farbe",8421504);
fb_Grdx=      pApp->GetProfileInt("Achsen","X_Grid_Farbe",65793*190);
fb_Grdy=      pApp->GetProfileInt("Achsen","Y_Grid_Farbe",65793*190);
linB_Grdx=    pApp->GetProfileInt("Achsen","X_Grid_Linienbreite",0);
linB_Grdy=    pApp->GetProfileInt("Achsen","Y_Grid_Linienbreite",0);
mod_Grdx=     pApp->GetProfileInt("Achsen","X_Grid_Modus",2);
mod_Grdy=     pApp->GetProfileInt("Achsen","Y_Grid_Modus",2);
```

```
sw_FK=        pApp->GetProfileInt("Funktion","Kurve",0);
linB_Fn=      pApp->GetProfileInt("Funktion","Kurvenlinienbreite",1);
mod_Fn=       pApp->GetProfileInt("Funktion","Kurvenmodus",1);
sw_xy =       pApp->GetProfileInt("Funktion","Punkte",1);
linB_FnP=     pApp->GetProfileInt("Funktion","Punktbreite",2);
mod_FnP=      pApp->GetProfileInt("Funktion","Punktmodus",1);
sw_inv=       pApp->GetProfileInt("Funktion","invers",0);
CI_Pp=   atof(pApp->GetProfileString("Funktion","Fp_CI","95.0"));
CI_Pr=   atof(pApp->GetProfileString("Funktion","rxy_CI","95.0"));
CI_Pe=   atof(pApp->GetProfileString("Funktion","Fe_CI","95.0"));
sw_pq=        pApp->GetProfileInt("Funktion","Fp_Theta",0);
p_e_sw=       pApp->GetProfileInt("Funktion","Epsilon_p",0);
p_p_sw=       pApp->GetProfileInt("Funktion","p_p",0);
filename=     pApp->GetProfileString("Funktion","Datei","GRP2.asc");
rxy_D=        pApp->GetProfileInt("Funktion","rxy_Darstellung_überschnitten",0);
rxy_=         pApp->GetProfileInt("Funktion","rxy_Darstellung",1);
ryx_=         pApp->GetProfileInt("Funktion","ryx_Darstellung",1);
sxy_=         pApp->GetProfileInt("Funktion","sxy_Darstellung",1);
syx_=         pApp->GetProfileInt("Funktion","syx_Darstellung",1);
s1xy_=        pApp->GetProfileInt("Funktion","s'xy_Darstellung",0);
s1yx_=        pApp->GetProfileInt("Funktion","s'yx_Darstellung",0);
srxy_=        pApp->GetProfileInt("Funktion","srxy_Darstellung",0);
sryx_=        pApp->GetProfileInt("Funktion","sryx_Darstellung",0);
srx_=         pApp->GetProfileInt("Funktion","srx_Darstellung",0);
sry_=         pApp->GetProfileInt("Funktion","sry_Darstellung",0);
am_=          pApp->GetProfileInt("Funktion","am_Darstellung",1);
sd_=          pApp->GetProfileInt("Funktion","sd_Darstellung",0);
sdg_=         pApp->GetProfileInt("Funktion","sd'_Darstellung",1);
sgam_=        pApp->GetProfileInt("Funktion","s'am_Darstellung",0);
a3_=          pApp->GetProfileInt("Funktion","a3_Darstellung",0);
a4_=          pApp->GetProfileInt("Funktion","a4_Darstellung",0);
ag3_=         pApp->GetProfileInt("Funktion","a3'_Darstellung",0);
ag4_=         pApp->GetProfileInt("Funktion","a4'_Darstellung",0);
sga3_=        pApp->GetProfileInt("Funktion","s'a3_Darstellung",1);
sga4_=        pApp->GetProfileInt("Funktion","s'a4_Darstellung",0);
e_=           pApp->GetProfileInt("Funktion","e_Darstellung",1);
x_=           pApp->GetProfileInt("Funktion","x_Darstellung",0);
x0_=          pApp->GetProfileInt("Funktion","x0_Darstellung",1);
x1_=          pApp->GetProfileInt("Funktion","x1_Darstellung",1);
s0_=          pApp->GetProfileInt("Funktion","s0_Darstellung",0);
s1_=          pApp->GetProfileInt("Funktion","s1_Darstellung",0);
emf_filename= pApp->GetProfileString("EMF","Datei","GRP_Diagramm_1.emf");
sc=      atof(pApp->GetProfileString("Dialog","Diagramm_Skalierung_1","1.2"));
scy=     atof(pApp->GetProfileString("Dialog","Diagramm_Skalierung_2","30"));
scrx=         pApp->GetProfileInt("Dialog","Emf_Skalierung_x",25);
scry=         pApp->GetProfileInt("Dialog","Emf_Skalierung_y",30);
coord.left=   pApp->GetProfileInt("Dialog","Position_x",579);
coord.top=    pApp->GetProfileInt("Dialog","Position_y",151);
coord.right=  pApp->GetProfileInt("Dialog","Grösse_x",399);
coord.bottom= pApp->GetProfileInt("Dialog","Grösse_y",268);
mv1x=    atof(pApp->GetProfileString("Dialog","Vektor_x","0.5"));
mv1y=    atof(pApp->GetProfileString("Dialog","Vektor_y","0.5"));
mv2x=    atof(pApp->GetProfileString("Dialog","Achse_x","0.5"));
mv2y=    atof(pApp->GetProfileString("Dialog","Achse_y","0.5"));
linB_Fn=      pApp->GetProfileInt("Funktion","Linienbreite",1);
//dt_xy=        pApp->GetProfileInt("Funktion","Punktgrösse",3);
fb_hg=        pApp->GetProfileInt("Diagramm","Hintergrundfarbe",13357270);
fb_K=         pApp->GetProfileInt("Funktion","Linienfarbe",0);
fb_P=         pApp->GetProfileInt("Funktion","Punktfarbe",0);
linB_Ax=      pApp->GetProfileInt("Achsen","x_Linienbreite",1);
linB_Ay=      pApp->GetProfileInt("Achsen","y_Linienbreite",1);
linB_Vx=      pApp->GetProfileInt("Achsen","x_Vektor_Linienbreite",1);
linB_Vy=      pApp->GetProfileInt("Achsen","y_Vektor_Linienbreite",1);
linB_r=       pApp->GetProfileInt("Funktion","rxy_Linienbreite",1);
linB_ri=      pApp->GetProfileInt("Funktion","ryx_Linienbreite",1);
linB_sr=      pApp->GetProfileInt("Funktion","srxy_Linienbreite",1);
linB_sri=     pApp->GetProfileInt("Funktion","sryx_Linienbreite",1);
linB_sR=      pApp->GetProfileInt("Funktion","sry_Linienbreite",1);
linB_sRi=     pApp->GetProfileInt("Funktion","srx_Linienbreite",1);
linB_sxy=     pApp->GetProfileInt("Funktion","sxy_Linienbreite",1);
linB_syx=     pApp->GetProfileInt("Funktion","syx_Linienbreite",1);
linB_sgxy=    pApp->GetProfileInt("Funktion","s'xy_Linienbreite",1);
linB_sgyx=    pApp->GetProfileInt("Funktion","s'yx_Linienbreite",1);
linB_am=      pApp->GetProfileInt("Funktion","am_Linienbreite",1);
linB_sd=      pApp->GetProfileInt("Funktion","sd_Linienbreite",1);
linB_sgam=    pApp->GetProfileInt("Funktion","s'am_Linienbreite",1);
linB_sdg=     pApp->GetProfileInt("Funktion","sd'_Linienbreite",1);
linB_a3=      pApp->GetProfileInt("Funktion","a3_Linienbreite",1);
```

```
linB_ag3=     pApp->GetProfileInt("Funktion","a3'_Linienbreite",1);
linB_sga3=    pApp->GetProfileInt("Funktion","s'a3_Linienbreite",1);
linB_a4=      pApp->GetProfileInt("Funktion","a4_Linienbreite",1);
linB_ag4=     pApp->GetProfileInt("Funktion","a4'_Linienbreite",1);
linB_sga4=    pApp->GetProfileInt("Funktion","s'a4_Linienbreite",1);
linB_e=       pApp->GetProfileInt("Funktion","e_Linienbreite",1);
linB_x=       pApp->GetProfileInt("Funktion","x_Linienbreite",1);
linB_x0=      pApp->GetProfileInt("Funktion","x0_Linienbreite",1);
linB_x1=      pApp->GetProfileInt("Funktion","x1_Linienbreite",1);
linB_s0=      pApp->GetProfileInt("Funktion","s0_Linienbreite",1);
linB_s1=      pApp->GetProfileInt("Funktion","s1_Linienbreite",1);
mod_Ax=       pApp->GetProfileInt("Achsen","X_Modus",1);
mod_Ay=       pApp->GetProfileInt("Achsen","Y_Modus",1);
mod_Vx=       pApp->GetProfileInt("Achsen","x_Vektor_Modus",2);
mod_Vy=       pApp->GetProfileInt("Achsen","y_Vektor_Modus",2);
mod_r=        pApp->GetProfileInt("Funktion","rxy_Modus",1);
mod_ri=       pApp->GetProfileInt("Funktion","ryx_Modus",1);
mod_sr=       pApp->GetProfileInt("Funktion","srxy_Modus",2);
mod_sri=      pApp->GetProfileInt("Funktion","sryx_Modus",2);
mod_sR=       pApp->GetProfileInt("Funktion","sry_Modus",2);
mod_sRi=      pApp->GetProfileInt("Funktion","srx_Modus",2);
mod_sxy=      pApp->GetProfileInt("Funktion","sxy_Modus",1);
mod_syx=      pApp->GetProfileInt("Funktion","syx_Modus",1);
mod_sgxy=     pApp->GetProfileInt("Funktion","s'xy_Modus",3);
mod_sgyx=     pApp->GetProfileInt("Funktion","s'yx_Modus",3);
mod_am=       pApp->GetProfileInt("Funktion","am_Modus",1);
mod_sd=       pApp->GetProfileInt("Funktion","sd_Modus",1);
mod_sgam=     pApp->GetProfileInt("Funktion","s'am_Modus",2);
mod_sdg=      pApp->GetProfileInt("Funktion","sd'_Modus",3);
mod_a3=       pApp->GetProfileInt("Funktion","a3_Modus",1);
mod_ag3=      pApp->GetProfileInt("Funktion","a3'_Modus",3);
mod_sga3=     pApp->GetProfileInt("Funktion","s'a3_Modus",2);
mod_a4=       pApp->GetProfileInt("Funktion","a4_Modus",1);
mod_ag4=      pApp->GetProfileInt("Funktion","a4'_Modus",3);
mod_sga4=     pApp->GetProfileInt("Funktion","s'a4_Modus",2);
mod_e=        pApp->GetProfileInt("Funktion","e_Modus",2);
mod_x=        pApp->GetProfileInt("Funktion","x_Modus",1);
mod_x0=       pApp->GetProfileInt("Funktion","x0_Modus",1);
mod_x1=       pApp->GetProfileInt("Funktion","x1_Modus",1);
mod_s0=       pApp->GetProfileInt("Funktion","s0_Modus",1);
mod_s1=       pApp->GetProfileInt("Funktion","s1_Modus",1);
r_q =         pApp->GetProfileInt("Funktion","rxy_Theta_Darstellung",1);
ri_q =        pApp->GetProfileInt("Funktion","ryx_Theta_Darstellung",1);
syx_q  =      pApp->GetProfileInt("Funktion","syx_Theta_Darstellung",1);
sxy_q  =      pApp->GetProfileInt("Funktion","sxy_Theta_Darstellung",1);
sgyx_q  =     pApp->GetProfileInt("Funktion","s'yx_Theta_Darstellung",1);
sgxy_q  =     pApp->GetProfileInt("Funktion","s'xy_Theta_Darstellung",1);
sr_q  =       pApp->GetProfileInt("Funktion","srxy_Theta_Darstellung",2);
sri_q  =      pApp->GetProfileInt("Funktion","sryx_Theta_Darstellung",2);
sR_q  =       pApp->GetProfileInt("Funktion","srx_Theta_Darstellung",0);
sRi_q =       pApp->GetProfileInt("Funktion","sry_Theta_Darstellung",0);
am_q  =       pApp->GetProfileInt("Funktion","am_Theta_Darstellung",1);
sd_q  =       pApp->GetProfileInt("Funktion","sd_Theta_Darstellung",1);
sdg_q  =      pApp->GetProfileInt("Funktion","sd'_Theta_Darstellung",1);
sgam_q  =     pApp->GetProfileInt("Funktion","s'am_Theta_Darstellung",2);
a3_q  =       pApp->GetProfileInt("Funktion","a3_Theta_Darstellung",2);
ag3_q  =      pApp->GetProfileInt("Funktion","a3'_Theta_Darstellung",2);
sa3g_q  =     pApp->GetProfileInt("Funktion","s'a3_Theta_Darstellung",1);
a4_q  =       pApp->GetProfileInt("Funktion","a4_Theta_Darstellung",2);
ag4_q  =      pApp->GetProfileInt("Funktion","a4'_Theta_Darstellung",2);
sa4g_q  =     pApp->GetProfileInt("Funktion","s'a4_Theta_Darstellung",1);
e_q  =        pApp->GetProfileInt("Funktion","e_Theta_Darstellung",1);
x_q  =        pApp->GetProfileInt("Funktion","x_Theta_Darstellung",1);
x0_q  =       pApp->GetProfileInt("Funktion","x0_Theta_Darstellung",1);
x1_q  =       pApp->GetProfileInt("Funktion","x1_Theta_Darstellung",1);
s0_q  =       pApp->GetProfileInt("Funktion","s0_Theta_Darstellung",1);
s1_q  =       pApp->GetProfileInt("Funktion","s1_Theta_Darstellung",1);
sw_emf_in=    pApp->GetProfileInt("EMF","öffnen",0);
tlg_x=   atof(pApp->GetProfileString("Achsen","x_Skala_Teilung","2"));
tlg_y=   atof(pApp->GetProfileString("Achsen","y_Skala_Teilung","2"));
fn_x_fon=     pApp->GetProfileString("Schriftart","Funktion_x","Arial");
fn_y_fon=     pApp->GetProfileString("Schriftart","Funktion_y","Arial");
fn_x_fb=      pApp->GetProfileInt("Schriftart","Farbe_Funktion_x",8421504);
fn_y_fb=      pApp->GetProfileInt("Schriftart","Farbe_Funktion_y",8421504);
fn_x_H=       pApp->GetProfileInt("Schriftart","Höhe_Funktion_x",13);
fn_y_H=       pApp->GetProfileInt("Schriftart","Höhe_Funktion_y",13);
fn_x_W=       pApp->GetProfileInt("Schriftart","Breite_Funktion_x",4);
fn_y_W=       pApp->GetProfileInt("Schriftart","Breite_Funktion_y",4);
```

```
        Ax_fon=       pApp->GetProfileString("Schriftart","Achsen_x","Arial");
        Ay_fon=       pApp->GetProfileString("Schriftart","Achsen_y","Arial");
        Ax_fb=        pApp->GetProfileInt("Schriftart","Farbe_Achsen_x",8421504);
        Ay_fb=        pApp->GetProfileInt("Schriftart","Farbe_Achsen_y",8421504);
        Ax_H=         pApp->GetProfileInt("Schriftart","Höhe_Achsen_x",13);
        Ay_H=         pApp->GetProfileInt("Schriftart","Höhe_Achsen_y",13);
        Ax_W=         pApp->GetProfileInt("Schriftart","Breite_Achsen_x",4);
        Ay_W=         pApp->GetProfileInt("Schriftart","Breite_Achsen_y",4);
        V_fon=        pApp->GetProfileString("Schriftart","Vektor","Arial");
        V_fb=         pApp->GetProfileInt("Schriftart","Farbe_Vektor",8421504);
        V_H=          pApp->GetProfileInt("Schriftart","Höhe_Vektor",13);
        V_W=          pApp->GetProfileInt("Schriftart","Breite_Vektor",4);
        dynrnd=       pApp->GetProfileInt("Programm","Dynamisch_rendern",0);
        filestr=      pApp->GetProfileInt("Programm","Filestream_rendern",0);
        wnd_pos=      pApp->GetProfileInt("Programm","Fensterposition_speichern",1);
        log_=         pApp->GetProfileInt("Programm","Logfile",0);
        csr_=         pApp->GetProfileInt("Programm","Cursor",1);
        sw_splash=    pApp->GetProfileInt("Programm","Splash",1);
        sw_Fxy=       pApp->GetProfileInt("Programm","Funktionsmatrixfenster",0);
        sw_Theta=     pApp->GetProfileInt("Programm","Thetafenster",0);
        sw_Log=       pApp->GetProfileInt("Programm","Logfenster",0);
        sw_Log=       pApp->GetProfileInt("Programm","Logfenster",0);
        sw_status=    pApp->GetProfileInt("Programm","Statusleiste",0);
        sw_As=        pApp->GetProfileInt("Programm","Achsen_Verschub",1);
        sw_Vs=        pApp->GetProfileInt("Programm","Vektoren_Verschub",1);
        modus_=       pApp->GetProfileInt("Programm","Startmodus",0);

                      pApp->WriteProfileInt("Achsen","x_Justierung",0);
                      pApp->WriteProfileInt("Achsen","y_Justierung",0);

        // filename="GRP2.asc"; //

        log_filename="~~tmp_Log_";
        log_filename+=itoc(time(0));
        log_filename+=".txt";

        if (fopen (filename, "r") == 0) //autogenerierte funktionsmatrixdatei
        {
               FILE *f;         filename="GRP2.asc";
                     f = fopen (filename, "w");

               fprintf( f,"1\t6\n");
               fprintf( f,"2\t3\n");
               fprintf( f,"3\t8\n");
               fprintf( f,"4\t2\n");
               fprintf( f,"5\t6\n");
               fprintf( f,"6\t3\n");
               fprintf( f,"7\t9\n");
               fprintf( f,"8\t4\n");
               fprintf( f,"9\t2\n");
               fprintf( f,"10\t9\n");

               fclose( f );
        }

        sw_FK=0; sw_xy=1; mod_FnP=1; linB_FnP=2;// f(x) voreinstellung: keine kurve,  xy punkte
rund //

        swli=1;
        sw_integral=0;
        qR=0;
        sw_emf=0;
        corx=0;
        cory=0;
        corx0=0;
        cory0=0;
        sc0x=0;//negativbereichskorrektur x
        sc0y=0;//negativbereichskorrektur
        csr_0=csr_;
        sc_0=sc;
        scy_0=scy;
        sw_x_0=sw_x;
        sw_y_0=sw_y;
        sw_xm_0=sw_xm;
        sw_ym_0=sw_ym;
        sw_xA_0=sw_xA;
        sw_yA_0=sw_yA;
        sw_xV_0=sw_xV;
```

```
sw_yV_0=sw_yV;
sw_xS_0=sw_xS;
sw_yS_0=sw_yS;
tlg_x_0=tlg_x;
tlg_y_0=tlg_y;
sw_xSw_0=sw_xSw;
sw_ySw_0=sw_ySw;
sw_xK_0=sw_xK;
sw_yK_0=sw_yK;
sw_FK_0=sw_FK;
sw_xy_0=sw_xy;
ds_xk_0=ds_xk;
ds_yk_0=ds_yk;
ds_xSw_0=ds_xSw;
ds_ySw_0=ds_ySw;
ds_x_0=ds_x;
ds_y_0=ds_y;
fb_hg_0=fb_hg;
fb_K_0=fb_K;
fb_P_0=fb_P;
linB_Fn_0=linB_Fn;
linB_FnP_0=linB_FnP;
fn_x_fon_0=fn_x_fon;
fn_x_fb_0=fn_x_fb;
fn_x_H_0=fn_x_H;
fn_x_W_0=fn_x_W;
fn_y_fon_0=fn_y_fon;
fn_y_fb_0=fn_y_fb;
fn_y_H_0=fn_y_H;
fn_y_W_0=fn_y_W;
Ax_fon_0=Ax_fon;
Ax_fb_0=Ax_fb;
Ax_H_0=Ax_H;
Ax_W_0=Ax_W;
Ay_fon_0=Ay_fon;
Ay_fb_0=Ay_fb;
Ay_H_0=Ay_H;
Ay_W_0=Ay_W;
V_fon_0=V_fon;
V_fb_0=V_fb;
V_H_0=V_H;
V_W_0=V_W;
mod_Fn_0=mod_Fn;
mod_FnP_0=mod_FnP;
mod_Ax_0=mod_Ax;
mod_Ay_0=mod_Ay;
mod_Vx_0=mod_Vx;
mod_Vy_0=mod_Vy;
fb_Grdx_0=fb_Grdx;
fb_Grdy_0=fb_Grdy;
linB_Grdx_0=linB_Grdx;
linB_Grdy_0=linB_Grdy;
mod_Grdx_0=mod_Grdx;
mod_Grdy_0=mod_Grdy;
mod_ri_0=mod_ri;
mod_sr_0=mod_sr;
mod_sri_0=mod_sri;
mod_sR_0=mod_sR;
mod_sRi_0=mod_sRi;
mod_sxy_0=mod_sxy;
mod_syx_0=mod_syx;
mod_sgxy_0=mod_sgxy;
mod_sgyx_0=mod_sgyx;
mod_am_0=mod_am;
mod_sd_0=mod_sd;
mod_sgam_0=mod_sgam;
mod_sdg_0=mod_sdg;
mod_a3_0=mod_a3;
mod_ag3_0=mod_ag3;
mod_sga3_0=mod_sga3;
mod_a4_0=mod_a4;
mod_ag4_0=mod_ag4;
mod_sga4_0=mod_sga4;
mod_e_0=mod_e;
mod_x_0=mod_x;
mod_x0_0=mod_x0;
mod_x1_0=mod_x1;
mod_s0_0=mod_s0;
```

```
mod_s1_0=mod_s1;
linB_Ax_0=linB_Ax;
linB_Ay_0=linB_Ay;
linB_Vx_0=linB_Vx;
linB_Vy_0=linB_Vy;
linB_ri_0=linB_ri;
linB_sr_0=linB_sr;
linB_sri_0=linB_sri;
linB_sR_0=linB_sR;
linB_sRi_0=linB_sRi;
linB_sxy_0=linB_sxy;
linB_syx_0=linB_syx;
linB_sgxy_0=linB_sgxy;
linB_sgyx_0=linB_sgyx;
linB_am_0=linB_am;
linB_sd_0=linB_sd;
linB_sgam_0=linB_sgam;
linB_sdg_0=linB_sdg;
linB_a3_0=linB_a3;
linB_ag3_0=linB_ag3;
linB_sga3_0=linB_sga3;
linB_a4_0=linB_a4;
linB_ag4_0=linB_ag4;
linB_sga4_0=linB_sga4;
linB_e_0=linB_e;
linB_x_0=linB_x;
linB_x0_0=linB_x0;
linB_x1_0=linB_x1;
linB_s0_0=linB_s0;
linB_s1_0=linB_s1;
fb_Ax_0=fb_Ax;
fb_Ay_0=fb_Ay;
fb_Vx_0=fb_Vx;
fb_Vy_0=fb_Vy;
fb_r_0=fb_r;
fb_ri_0=fb_ri;
fb_sr_0=fb_sr;
fb_sri_0=fb_sri;
fb_sR_0=fb_sR;
fb_sRi_0=fb_sRi;
fb_sxy_0=fb_sxy;
fb_syx_0=fb_syx;
fb_sgxy_0=fb_sgxy;
fb_sgyx_0=fb_sgyx;
fb_am_0=fb_am;
fb_sd_0=fb_sd;
fb_sgam_0=fb_sgam;
fb_sdg_0=fb_sdg;
fb_a3_0=fb_a3;
fb_ag3_0=fb_ag3;
fb_sga3_0=fb_sga3;
fb_a4_0=fb_a4;
fb_ag4_0=fb_ag4;
fb_sga4_0=fb_sga4;
fb_e_0=fb_e;
fb_x_0=fb_x;
fb_x0_0=fb_x0;
fb_x1_0=fb_x1;
fb_s0_0=fb_s0;
fb_s1_0=fb_s1;
rxy_D_0=rxy_D;
rxy_0=rxy_;
ryx_0=ryx_;
sxy_0=sxy_;
syx_0=syx_;
s1xy_0=s1xy_;
s1yx_0=s1yx_;
srxy_0=srxy_;
sryx_0=sryx_;
srx_0=srx_;
sry_0=sry_;
CI_Pp_0=CI_Pp;
CI_Pr_0=CI_Pr;
CI_Pe_0=CI_Pe;
ci_ze_0=ci_ze=0;
a3_0 = a3_; //a3 linienschalter
a4_0 = a4_;   //a4 linienschalter
ag3_0 = ag3_;   //a'3 linienschalter
```

```
ag4_0 = ag4_;    //a'4 linienschalter
am_0 = am_;   //am linienschalter
sd_0 = sd_;   //sd linienschalter
sdg_0 = sdg_;    //sd' linienschalter
sga3_0 = sga3_;   //s'a3 linienschalter
sga4_0 = sga4_; //s'a4 linienschalter
sgam_0 = sgam_; //s'am linienschalter
sw_pq_0=sw_pq;
sw_Grdx_0=sw_Grdx;
sw_Grdy_0=sw_Grdy;
posVx_0=posVx;
posVy_0=posVy;
posAXx_0=posAXx;
posAXy_0=posAXy;
posAYx_0=posAYx;
posAYy_0=posAYy;
posBXx_0=posBXx;
posBXy_0=posBXy;
posBYx_0=posBYx;
posBYy_0=posBYy;
posFXx_0=posFXx;
posFXy_0=posFXy;
posFYx_0=posFYx;
posFYy_0=posFYy;
posX_0=posX=0; //Diagramm x position
posY_0=posY=0; //Diagramm y position
frmX_0=frmX=1; //Diagramm x form
frmY_0=frmY=1; //Diagramm y form
r_q_0=r_q;
ri_q_0=ri_q;
syx_q_0=syx_q ;
sxy_q_0=sxy_q ;
sgyx_q_0=sgyx_q ;
sgxy_q_0=sgxy_q ;
sr_q_0=sr_q ;
sri_q_0=sri_q ;
sR_q_0=sR_q ;
sRi_q_0=sRi_q;
am_q_0=am_q ;
sd_q_0=sd_q ;
sdg_q_0=sdg_q ;
sgam_q_0=sgam_q ;
a3_q_0=a3_q ;
ag3_q_0=ag3_q ;
sa3g_q_0=sa3g_q ;
a4_q_0=a4_q ;
ag4_q_0=ag4_q ;
sa4g_q_0=sa4g_q ;
e_q_0=e_q;
x_q_0=x_q;
x0_q_0=x0_q;
x1_q_0=x1_q;
s0_q_0=s0_q;
s1_q_0=s1_q;
sw_xb_0=sw_xb;
sw_yb_0=sw_yb;
p_e_sw_0=p_e_sw;
p_p_sw_0=p_p_sw;

sw_v0=0; //vektorursprungspositionsschalter
xBz="";
xBz_0="";
yBz="";
yBz_0="";

SetTimer(0,50,0);  //haupt ereignisse
SetTimer(1,300,0); //rerender ereignis
SetTimer(2,10,0);  //rerender sizemarker switch ereignis

filename_= "~~tmp_.asc";
filename_z="~~tmp_z.asc";
filename_p="~~tmp_p.asc";

sw_mod_=1;//funktionsmodus

MINMAX();// theta0

sw_mkoord_A=1;mAx=min_x; mAy=min_y;   // achsenposition allgemein
```

```
        sw_mkoord_V=1;mVx=max_x; mVy=max_y;   // vektorposition

        mAx_0=mAx;
        mAy_0=mAy;
        mVx_0=mVx;
        mVy_0=mVy;

        fn_THETA_1(); // theta desc

                         if(fil!="")filename=fil;  //cmdline (weitere argumente zu 
bearbeiten...)
                        Fenstertext(filename);     //funktionsmatrixmodus start
        if(sw_emf_in==1)Fenstertext(emf_filename); //emf start

        filename_tmp=filename;

        if(log_)log_file(1); //Funktionsmatrix log;

        SetWindowPos(&wndTop, coord.left,coord.top, coord.right+8,coord.bottom+45, 
SWP_SHOWWINDOW); //hauptfenster position

        ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
        ASSERT(IDM_ABOUTBOX < 0xF000);

        CMenu* pSysMenu = GetSystemMenu(0); //system menue
        if (pSysMenu != NULL)
        {
                CString strAboutMenu;
                strAboutMenu.LoadString(IDS_ABOUTBOX);
                if (!strAboutMenu.IsEmpty())
                {
                        pSysMenu->AppendMenu(MF_SEPARATOR);
                        pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
                }
        }

        if(sw_emf_in!=1) //bei funktionsmatrixmodus start
        {
                CMenu        o;
                                      o.LoadMenu(IDR_MENU1);
                                      o.CheckMenuItem(ID_MODUS_FX, MF_CHECKED);
                if(sw_csr==2)o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_SCHWARZ,  MF_CHECKED);
                if(sw_csr==1)o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_WEISS,  MF_CHECKED);
                if(sw_As==1) o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_ACHSENVERSCHUB, 
MF_CHECKED);
                if(sw_Vs==1) o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_VEKTORENVERSCHUB, 
MF_CHECKED);
                if(sw_status)o.CheckMenuItem(ID_ANSICHT_STATUSLEISTE, MF_CHECKED);

                SetMenu(    &o); //haupt menue
        }

        SetIcon(m_hIcon, 1);SetIcon(m_hIcon, 0);

        if(sw_splash){GRP2splash sp; sp.DoModal();}//splash

        if(sw_emf_in!=1) //bei funktionsmatrixmodus start
        {
                if(sw_Log){sw_Log=0;OnAnsichtGrp2log(0);} //Log Ansicht (dazu menu modifikation 
dort)

                if(sw_Fxy){sw_Fxy=0;OnAnsichtFunktionsmatrixfxy(0);} //Fxy Ansicht (dazu menu 
modifikation dort)

                if(sw_Theta)//Theta Ansicht (dazu menu modifikation dort)
                {
                        sw_Theta=0;OnAnsichtThetafensterq(0);

                        //if(sw_Fxy)m_ThetaDlg.SetWindowPos(&wndTop, coord.left+30,coord.top+70, 
0,0, SWP_NOSIZE); //Thetafensterposition 0
                }

                //Startmodus

                //if(0)OnModusFx(); //automatisch
                if(modus_==5){sw_mod_=0;OnModusFzx();}
                if(modus_==2){sw_mod_=0;OnModusRxy();}
                if(modus_==3){sw_mod_=0;OnModusFp();}
```

```
               if(modus_==4){sw_mod_=0;OnModusEpsilon();}
        }

        if(sw_emf_in==1) //bei emf modus start
        {
                  CMenu o;
                               o.LoadMenu(IDR_MENU2);

           SetMenu(&o);//emf menue
        }

        return 1;
}

void CGRP2Dlg::OnSysCommand(UINT nID, LPARAM lParam)
{
        if ((nID & 0xFFF0) == IDM_ABOUTBOX)
        {
                m_InfoDlg.DestroyWindow();
                m_InfoDlg.Create(IDD_ABOUTBOX );
                m_InfoDlg.ShowWindow(SW_SHOW);
                m_InfoDlg.BringWindowToTop();
        }
        else{CDialog::OnSysCommand(nID, lParam);}
}


void CGRP2Dlg::OnPaint()
{
        CPaintDC ooo(this);

        if(sw_emf==1) { GRP_Diagramm();sw_emf=0;} //emf erstellen

        if(sw_emf_in==1) // emf darstellen
        {
                                                            CRect emf(0, 0,dlg.x, dlg.y);
                PlayEnhMetaFile(ooo, GetEnhMetaFile(emf_filename), emf);
        }


        if(sw_emf!=1)if(sw_emf_in!=1)
        {

        ///////////////////////////////////////////////////////////////////////////////////////////////
///////////

                if(fb_hg!=13357270) //hintergrundfarbe nicht windowsgrau
                {
                        //UpdateWindow();
                                                CRect rect(0, 0,dlg.x, dlg.y);
                        ooo.FillSolidRect(rect,fb_hg);
                }

                // schriftartendefinition
                CFont of1;of1.CreateFont(Ay_H,  Ay_W,
0,0,400,0,0,0,0,OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,DEFAULT_PITCH,Ay_fon
);//y Achse
                CFont of2;of2.CreateFont(Ax_H,  Ax_W,
0,0,400,0,0,0,0,OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,DEFAULT_PITCH,Ax_fon
);//x Achse
            CFont
ofx;ofx.CreateFont(fn_x_H,fn_x_W,0,0,400,0,0,0,0,OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAUL
T_QUALITY,DEFAULT_PITCH,fn_x_fon);//x funktionswerte
                CFont
ofy;ofy.CreateFont(fn_y_H,fn_y_W,0,0,400,0,0,0,0,OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAUL
T_QUALITY,DEFAULT_PITCH,fn_y_fon);//y funktionswerte
                CFont ofv;ofv.CreateFont(V_H,   V_W,
0,0,400,0,0,0,0,OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,DEFAULT_PITCH,V_fon
);//xy Vektor

                //linienartdefinition
                               CPen oos;
                if(mod_Ay==1)oos.CreatePen(PS_SOLID,     linB_Ay,fb_Ay); //y achse
                if(mod_Ay==2)oos.CreatePen(PS_DOT,       linB_Ay,fb_Ay); //...
                if(mod_Ay==3)oos.CreatePen(PS_DASH,      linB_Ay,fb_Ay); //...
                if(mod_Ay==4)oos.CreatePen(PS_DASHDOT,   linB_Ay,fb_Ay); //...
                if(mod_Ay==5)oos.CreatePen(PS_DASHDOTDOT,linB_Ay,fb_Ay); //...
                        CPen o1s;
                if(mod_Ax==1)o1s.CreatePen(PS_SOLID,     linB_Ax,fb_Ax); //x achse
                if(mod_Ax==2)o1s.CreatePen(PS_DOT,       linB_Ax,fb_Ax); //...
                if(mod_Ax==3)o1s.CreatePen(PS_DASH,      linB_Ax,fb_Ax); //...
                if(mod_Ax==4)o1s.CreatePen(PS_DASHDOT,   linB_Ax,fb_Ax); //...
                if(mod_Ax==5)o1s.CreatePen(PS_DASHDOTDOT,linB_Ax,fb_Ax); //...
                        CPen o3s;
                             o3s.CreatePen(PS_SOLID,     linB_Ax,fb_Ax); //x achsen
teilungsstriche
                        CPen o4s;
                                    o4s.CreatePen(PS_SOLID,     linB_Ay,fb_Ay); //y achsen
teilungsstriche
                        CPen o2s;
                if(mod_Fn==1)o2s.CreatePen(PS_SOLID,     linB_Fn,fb_K); //funktionskurve
                if(mod_Fn==2)o2s.CreatePen(PS_DOT,       linB_Fn,fb_K);
                if(mod_Fn==3)o2s.CreatePen(PS_DASH,      linB_Fn,fb_K);
                if(mod_Fn==4)o2s.CreatePen(PS_DASHDOT,   linB_Fn,fb_K);
                if(mod_Fn==5)o2s.CreatePen(PS_DASHDOTDOT,linB_Fn,fb_K);
                        CPen o2s1;
                                o2s1.CreatePen(PS_SOLID,    linB_Fn,fb_P); //funktionspunkte
                        CPen o1;
         if(mod_Vx==1)o1.CreatePen(PS_SOLID,      linB_Vx,fb_Vx); //x vektor
                if(mod_Vx==2)o1.CreatePen(PS_DOT,        linB_Vx,fb_Vx); //...
                if(mod_Vx==3)o1.CreatePen(PS_DASH,       linB_Vx,fb_Vx); //...
                if(mod_Vx==4)o1.CreatePen(PS_DASHDOT,    linB_Vx,fb_Vx); //...
                if(mod_Vx==5)o1.CreatePen(PS_DASHDOTDOT, linB_Vx,fb_Vx); //...
                        CPen o2;
         if(mod_Vy==1)o2.CreatePen(PS_SOLID,      linB_Vy,fb_Vy); //y vektor
                if(mod_Vy==2)o2.CreatePen(PS_DOT,        linB_Vy,fb_Vy); //...
                if(mod_Vy==3)o2.CreatePen(PS_DASH,       linB_Vy,fb_Vy); //...
                if(mod_Vy==4)o2.CreatePen(PS_DASHDOT,    linB_Vy,fb_Vy); //...
                if(mod_Vy==5)o2.CreatePen(PS_DASHDOTDOT, linB_Vy,fb_Vy); //...
                        CPen or1;
                if(mod_r==1)or1.CreatePen(PS_SOLID,     linB_r,fb_r); //r
                if(mod_r==2)or1.CreatePen(PS_DOT,       linB_r,fb_r); //...
                if(mod_r==3)or1.CreatePen(PS_DASH,      linB_r,fb_r); //...
                if(mod_r==4)or1.CreatePen(PS_DASHDOT,   linB_r,fb_r); //...
                if(mod_r==5)or1.CreatePen(PS_DASHDOTDOT,linB_r,fb_r); //...
                           CPen or1i;
                if(mod_ri==1)or1i.CreatePen(PS_SOLID,     linB_ri,fb_ri); //ri
                if(mod_ri==2)or1i.CreatePen(PS_DOT,       linB_ri,fb_ri); //...
                if(mod_ri==3)or1i.CreatePen(PS_DASH,      linB_ri,fb_ri); //...
                if(mod_ri==4)or1i.CreatePen(PS_DASHDOT,   linB_ri,fb_ri); //...
                if(mod_ri==5)or1i.CreatePen(PS_DASHDOTDOT,linB_ri,fb_ri); //...
                               CPen or2;
                if(mod_sr==1)or2.CreatePen(PS_SOLID,     linB_sr,fb_sr); //sr
                if(mod_sr==2)or2.CreatePen(PS_DOT,       linB_sr,fb_sr); //...
                if(mod_sr==3)or2.CreatePen(PS_DASH,      linB_sr,fb_sr); //...
                if(mod_sr==4)or2.CreatePen(PS_DASHDOT,   linB_sr,fb_sr); //...
                if(mod_sr==5)or2.CreatePen(PS_DASHDOTDOT,linB_sr,fb_sr); //...
                           CPen or2i;
                if(mod_sri==1)or2i.CreatePen(PS_SOLID,     linB_sri,fb_sri); //sri
                if(mod_sri==2)or2i.CreatePen(PS_DOT,       linB_sri,fb_sri); //...
                if(mod_sri==3)or2i.CreatePen(PS_DASH,      linB_sri,fb_sri); //...
                if(mod_sri==4)or2i.CreatePen(PS_DASHDOT,   linB_sri,fb_sri); //...
                if(mod_sri==5)or2i.CreatePen(PS_DASHDOTDOT,linB_sri,fb_sri); //...
                               CPen or3;
                if(mod_sR==1)or3.CreatePen(PS_SOLID,     linB_sR,fb_sR); //sR
                if(mod_sR==2)or3.CreatePen(PS_DOT,       linB_sR,fb_sR); //...
                if(mod_sR==3)or3.CreatePen(PS_DASH,      linB_sR,fb_sR); //...
                if(mod_sR==4)or3.CreatePen(PS_DASHDOT,   linB_sR,fb_sR); //...
                if(mod_sR==5)or3.CreatePen(PS_DASHDOTDOT,linB_sR,fb_sR); //...
                           CPen or3i;
                if(mod_sRi==1)or3i.CreatePen(PS_SOLID,     linB_sRi,fb_sRi); //sRi
                if(mod_sRi==2)or3i.CreatePen(PS_DOT,       linB_sRi,fb_sRi); //...
                if(mod_sRi==3)or3i.CreatePen(PS_DASH,      linB_sRi,fb_sRi); //...
                if(mod_sRi==4)or3i.CreatePen(PS_DASHDOT,   linB_sRi,fb_sRi); //...
                if(mod_sRi==5)or3i.CreatePen(PS_DASHDOTDOT,linB_sRi,fb_sRi); //...
                               CPen or4i;
                if(mod_sxy==1)or4i.CreatePen(PS_SOLID,     linB_sxy,fb_sxy); //sxy
                if(mod_sxy==2)or4i.CreatePen(PS_DOT,       linB_sxy,fb_sxy); //...
                if(mod_sxy==3)or4i.CreatePen(PS_DASH,      linB_sxy,fb_sxy); //...
                if(mod_sxy==4)or4i.CreatePen(PS_DASHDOT,   linB_sxy,fb_sxy); //...
                if(mod_sxy==5)or4i.CreatePen(PS_DASHDOTDOT,linB_sxy,fb_sxy); //...
                           CPen or4;
                if(mod_syx==1)or4.CreatePen(PS_SOLID,     linB_syx,fb_syx); //syx
                if(mod_syx==2)or4.CreatePen(PS_DOT,       linB_syx,fb_syx); //...
                if(mod_syx==3)or4.CreatePen(PS_DASH,      linB_syx,fb_syx); //...
                if(mod_syx==4)or4.CreatePen(PS_DASHDOT,   linB_syx,fb_syx); //...
                if(mod_syx==5)or4.CreatePen(PS_DASHDOTDOT,linB_syx,fb_syx); //...
                             CPen or5i;
                if(mod_sgxy==1)or5i.CreatePen(PS_SOLID,     linB_sgxy,fb_sgxy); //s'xy
                if(mod_sgxy==2)or5i.CreatePen(PS_DOT,       linB_sgxy,fb_sgxy); //...
                if(mod_sgxy==3)or5i.CreatePen(PS_DASH,      linB_sgxy,fb_sgxy); //...
                if(mod_sgxy==4)or5i.CreatePen(PS_DASHDOT,   linB_sgxy,fb_sgxy); //...
                if(mod_sgxy==5)or5i.CreatePen(PS_DASHDOTDOT,linB_sgxy,fb_sgxy); //...
                               CPen or5;
                if(mod_sgyx==1)or5.CreatePen(PS_SOLID,     linB_sgyx,fb_sgyx); //s'yx
                if(mod_sgyx==2)or5.CreatePen(PS_DOT,       linB_sgyx,fb_sgyx); //...
                if(mod_sgyx==3)or5.CreatePen(PS_DASH,      linB_sgyx,fb_sgyx); //...
                if(mod_sgyx==4)or5.CreatePen(PS_DASHDOT,   linB_sgyx,fb_sgyx); //...
                if(mod_sgyx==5)or5.CreatePen(PS_DASHDOTDOT,linB_sgyx,fb_sgyx); //...
                        CPen op1;
                if(mod_am==1)op1.CreatePen(PS_SOLID,     linB_am,fb_am); //am
                if(mod_am==2)op1.CreatePen(PS_DOT,       linB_am,fb_am); //...
                if(mod_am==3)op1.CreatePen(PS_DASH,      linB_am,fb_am); //...
                if(mod_am==4)op1.CreatePen(PS_DASHDOT,   linB_am,fb_am); //...
                if(mod_am==5)op1.CreatePen(PS_DASHDOTDOT,linB_am,fb_am); //...
                               CPen op2;
                if(mod_sd==1)op2.CreatePen(PS_SOLID,     linB_sd,fb_sd); //sd
                if(mod_sd==2)op2.CreatePen(PS_DOT,       linB_sd,fb_sd); //...
                if(mod_sd==3)op2.CreatePen(PS_DASH,      linB_sd,fb_sd); //...
                if(mod_sd==4)op2.CreatePen(PS_DASHDOT,   linB_sd,fb_sd); //...
                if(mod_sd==5)op2.CreatePen(PS_DASHDOTDOT,linB_sd,fb_sd); //...
                           CPen op3;
                if(mod_sgam==1)op3.CreatePen(PS_SOLID,     linB_sgam,fb_sgam); //s'am
                if(mod_sgam==2)op3.CreatePen(PS_DOT,       linB_sgam,fb_sgam); //...
                if(mod_sgam==3)op3.CreatePen(PS_DASH,      linB_sgam,fb_sgam); //...
                if(mod_sgam==4)op3.CreatePen(PS_DASHDOT,   linB_sgam,fb_sgam); //...
                if(mod_sgam==5)op3.CreatePen(PS_DASHDOTDOT,linB_sgam,fb_sgam); //...
                        CPen op4;
                if(mod_sdg==1)op4.CreatePen(PS_SOLID,     linB_sdg,fb_sdg); //sd'
                if(mod_sdg==2)op4.CreatePen(PS_DOT,       linB_sdg,fb_sdg); //...
                if(mod_sdg==3)op4.CreatePen(PS_DASH,      linB_sdg,fb_sdg); //...
                if(mod_sdg==4)op4.CreatePen(PS_DASHDOT,   linB_sdg,fb_sdg); //...
                if(mod_sdg==5)op4.CreatePen(PS_DASHDOTDOT,linB_sdg,fb_sdg); //...
                        CPen op5;
                if(mod_a3==1)op5.CreatePen(PS_SOLID,     linB_a3,fb_a3); //a3
                if(mod_a3==2)op5.CreatePen(PS_DOT,       linB_a3,fb_a3); //...
                if(mod_a3==3)op5.CreatePen(PS_DASH,      linB_a3,fb_a3); //...
                if(mod_a3==4)op5.CreatePen(PS_DASHDOT,   linB_a3,fb_a3); //...
                if(mod_a3==5)op5.CreatePen(PS_DASHDOTDOT,linB_a3,fb_a3); //...
                        CPen op6;
                if(mod_ag3==1)op6.CreatePen(PS_SOLID,     linB_ag3,fb_ag3); //a3'
                if(mod_ag3==2)op6.CreatePen(PS_DOT,       linB_ag3,fb_ag3); //...
                if(mod_ag3==3)op6.CreatePen(PS_DASH,      linB_ag3,fb_ag3); //...
                if(mod_ag3==4)op6.CreatePen(PS_DASHDOT,   linB_ag3,fb_ag3); //...
                if(mod_ag3==5)op6.CreatePen(PS_DASHDOTDOT,linB_ag3,fb_ag3); //...
                        CPen op7;
                if(mod_sga3==1)op7.CreatePen(PS_SOLID,     linB_sga3,fb_sga3); //s'a3
                if(mod_sga3==2)op7.CreatePen(PS_DOT,       linB_sga3,fb_sga3); //...
                if(mod_sga3==3)op7.CreatePen(PS_DASH,      linB_sga3,fb_sga3); //...
                if(mod_sga3==4)op7.CreatePen(PS_DASHDOT,   linB_sga3,fb_sga3); //...
                if(mod_sga3==5)op7.CreatePen(PS_DASHDOTDOT,linB_sga3,fb_sga3); //...
                        CPen op8;
                if(mod_a4==1)op8.CreatePen(PS_SOLID,     linB_a4,fb_a4); //a4
                if(mod_a4==2)op8.CreatePen(PS_DOT,       linB_a4,fb_a4); //...
                if(mod_a4==3)op8.CreatePen(PS_DASH,      linB_a4,fb_a4); //...
                if(mod_a4==4)op8.CreatePen(PS_DASHDOT,   linB_a4,fb_a4); //...
                if(mod_a4==5)op8.CreatePen(PS_DASHDOTDOT,linB_a4,fb_a4); //...
                        CPen op9;
                if(mod_ag4==1)op9.CreatePen(PS_SOLID,     linB_ag4,fb_ag4); //a4'
                if(mod_ag4==2)op9.CreatePen(PS_DOT,       linB_ag4,fb_ag4); //...
                if(mod_ag4==3)op9.CreatePen(PS_DASH,      linB_ag4,fb_ag4); //...
                if(mod_ag4==4)op9.CreatePen(PS_DASHDOT,   linB_ag4,fb_ag4); //...
                if(mod_ag4==5)op9.CreatePen(PS_DASHDOTDOT,linB_ag4,fb_ag4); //...
                        CPen op10;
                if(mod_sga4==1)op10.CreatePen(PS_SOLID,     linB_sga4,fb_sga4); //s'a4
                if(mod_sga4==2)op10.CreatePen(PS_DOT,       linB_sga4,fb_sga4); //...
                if(mod_sga4==3)op10.CreatePen(PS_DASH,      linB_sga4,fb_sga4); //...
                if(mod_sga4==4)op10.CreatePen(PS_DASHDOT,   linB_sga4,fb_sga4); //...
                if(mod_sga4==5)op10.CreatePen(PS_DASHDOTDOT,linB_sga4,fb_sga4); //...
                       CPen op11;
                if(mod_e==1)op11.CreatePen(PS_SOLID,     linB_e,fb_e); //e
                if(mod_e==2)op11.CreatePen(PS_DOT,       linB_e,fb_e); //...
                if(mod_e==3)op11.CreatePen(PS_DASH,      linB_e,fb_e); //...
                if(mod_e==4)op11.CreatePen(PS_DASHDOT,   linB_e,fb_e); //...
```

```
               if(mod_e==5)op11.CreatePen(PS_DASHDOTDOT,linB_e,fb_e); //...
                      CPen op12;
               if(mod_x==1)op12.CreatePen(PS_SOLID,     linB_x,fb_x); //x
               if(mod_x==2)op12.CreatePen(PS_DOT,       linB_x,fb_x); //...
               if(mod_x==3)op12.CreatePen(PS_DASH,      linB_x,fb_x); //...
               if(mod_x==4)op12.CreatePen(PS_DASHDOT,   linB_x,fb_x); //...
               if(mod_x==5)op12.CreatePen(PS_DASHDOTDOT,linB_x,fb_x); //...
                       CPen op13;
               if(mod_x0==1)op13.CreatePen(PS_SOLID,     linB_x0,fb_x0); //x0
               if(mod_x0==2)op13.CreatePen(PS_DOT,       linB_x0,fb_x0); //...
               if(mod_x0==3)op13.CreatePen(PS_DASH,      linB_x0,fb_x0); //...
               if(mod_x0==4)op13.CreatePen(PS_DASHDOT,   linB_x0,fb_x0); //...
               if(mod_x0==5)op13.CreatePen(PS_DASHDOTDOT,linB_x0,fb_x0); //...
                       CPen op14;
               if(mod_x1==1)op14.CreatePen(PS_SOLID,     linB_x1,fb_x1); //x1
               if(mod_x1==2)op14.CreatePen(PS_DOT,       linB_x1,fb_x1); //...
               if(mod_x1==3)op14.CreatePen(PS_DASH,      linB_x1,fb_x1); //...
               if(mod_x1==4)op14.CreatePen(PS_DASHDOT,   linB_x1,fb_x1); //...
               if(mod_x1==5)op14.CreatePen(PS_DASHDOTDOT,linB_x1,fb_x1); //...
                       CPen op15;
               if(mod_s0==1)op15.CreatePen(PS_SOLID,     linB_s0,fb_s0); //s0
               if(mod_s0==2)op15.CreatePen(PS_DOT,       linB_s0,fb_s0); //...
               if(mod_s0==3)op15.CreatePen(PS_DASH,      linB_s0,fb_s0); //...
               if(mod_s0==4)op15.CreatePen(PS_DASHDOT,   linB_s0,fb_s0); //...
               if(mod_s0==5)op15.CreatePen(PS_DASHDOTDOT,linB_s0,fb_s0); //...
                       CPen op16;
               if(mod_s1==1)op16.CreatePen(PS_SOLID,     linB_s1,fb_s1); //s1
               if(mod_s1==2)op16.CreatePen(PS_DOT,       linB_s1,fb_s1); //...
               if(mod_s1==3)op16.CreatePen(PS_DASH,      linB_s1,fb_s1); //...
               if(mod_s1==4)op16.CreatePen(PS_DASHDOT,   linB_s1,fb_s1); //...
               if(mod_s1==5)op16.CreatePen(PS_DASHDOTDOT,linB_s1,fb_s1); //...
                                CPen ogx;
               if(mod_Grdx==1)ogx.CreatePen(PS_SOLID,     linB_Grdx,fb_Grdx); //gridx
               if(mod_Grdx==2)ogx.CreatePen(PS_DOT,       linB_Grdx,fb_Grdx); //...
               if(mod_Grdx==3)ogx.CreatePen(PS_DASH,      linB_Grdx,fb_Grdx); //...
               if(mod_Grdx==4)ogx.CreatePen(PS_DASHDOT,   linB_Grdx,fb_Grdx); //...
               if(mod_Grdx==5)ogx.CreatePen(PS_DASHDOTDOT,linB_Grdx,fb_Grdx); //...
                                CPen ogy;
               if(mod_Grdy==1)ogy.CreatePen(PS_SOLID,     linB_Grdy,fb_Grdy); //gridy
               if(mod_Grdy==2)ogy.CreatePen(PS_DOT,       linB_Grdy,fb_Grdy); //...
               if(mod_Grdy==3)ogy.CreatePen(PS_DASH,      linB_Grdy,fb_Grdy); //...
               if(mod_Grdy==4)ogy.CreatePen(PS_DASHDOT,   linB_Grdy,fb_Grdy); //...
               if(mod_Grdy==5)ogy.CreatePen(PS_DASHDOTDOT,linB_Grdy,fb_Grdy); //...


               //füllwerkzeugdefinition
               CBrush b1;b1.CreateSolidBrush(fb_hg);//xy punkt füllfarbe hintergrund
               CBrush b2;b2.CreateSolidBrush(fb_P );//xy punkt rechteck rahmenfarbe

               //rendering

               ooo.SelectObject(&oos);
               ooo.SelectObject(&of1);
               ooo.SetBkColor( fb_hg);//hintergrundfarbe
               ooo.SetTextColor(Ay_fb);
            //ooo.SetBkMode(TRANSPARENT);

                       float dx_s =  dlg.x/sc;                // skalierte dialoggrösse x

                       float dx_sy = dlg.x/scy;               // skalierte dialoggrösse x

                       float dy_s =  dlg.y/sc;                // skalierte dialoggrösse y

                       float dy_sy = dlg.y/scy;               // skalierte dialoggrösse y

                       //float e_x=1; // einheit x=1

                       float e_x=((min_x+sc0x)/(max_x+sc0x)); // einheit x

//float e_y=1; // einheit y=1

                       float e_y=((min_y+sc0y)/(max_y+sc0y)); // einheit y

                       float egx= e_x * dx_s ;                // gewichtete einheit x

                       float egy= e_y * dy_s ;                // gewichtete einheit y
```

```
                                   float daptx = egx *(mv2x/(e_x/sc)) ; //dialog-achsen-
koordinatenpunkt x
                 if(sw_mkoord_A) daptx = (((((mAx))  +sc0x)/(max_x+sc0x))* dx_s)+dx_sy;
//dialog-Achsen-koordinatenpunkt x manuell
                                                 //~~~
                                      Ax_m= (((( daptx)-dx_sy)/dx_s)*(max_x+sc0x))-sc0x;;
//global für koordinatenübergabe an koordinateneinstellungsdialog
                 if(!sw_mkoord_A)mAx=Ax_m;

                         if(sw_yA==1)
                     ooo.MoveTo( daptx*frmX+ posX, 0);         //y achse
                         if(sw_yA==1)
                         ooo.LineTo( daptx*frmX+ posX, dlg.y);     //

                                                                                         CString
ct_="y";
                                          if(sw_inv==1)
ct_="x";     //f-1(x)
                                                         if(sw_mod_==2)
ct_="z(y)";  //rxy
                                                         if(sw_mod_==3)if(sw_pq==0)
ct_="x=q";   //Fp
                                                         if(sw_mod_==3)if(sw_pq==1)
ct_="y=q";   //

if(sw_mod_==3)if(sw_inv==1)if(p_p_sw==0)if(sw_pq==0)ct_="p(x)";  //

if(sw_mod_==3)if(sw_inv==1)if(p_p_sw==0)if(sw_pq==1)ct_="p(y)";  //

if(sw_mod_==3)if(sw_inv==1)if(p_p_sw==1)if(sw_pq==0)ct_="pa1(x)";//

if(sw_mod_==3)if(sw_inv==1)if(p_p_sw==1)if(sw_pq==1)ct_="pa1(y)";//

if(sw_mod_==3)if(sw_inv==1)if(p_p_sw==2)if(sw_pq==0)ct_="pa2(x)";//

if(sw_mod_==3)if(sw_inv==1)if(p_p_sw==2)if(sw_pq==1)ct_="pa2(y)";//
                                                         if(sw_mod_==4)if(sw_inv==0)if(p_e_sw==0)
ct_="p";     //Fe
                                                         if(sw_mod_==4)if(sw_inv==0)if(p_e_sw==1)
ct_="pa1";   //
                                                         if(sw_mod_==4)if(sw_inv==0)if(p_e_sw==2)
ct_="pa2";   //
                                                         if(sw_mod_==4)if(sw_inv==1)
ct_="q";     //

                                                         if(sw_ybz)
ct_=yBz;    //manuell

                         if(sw_yA==1)
                         if(sw_yb>=1)ooo.TextOut(daptx*frmX-2+posBYx+ posX,        0+posBYy ,
ct_); //y achsen bezeichnung oben
                         if(sw_yA==1)
                         if(!sw_status)if(sw_yb==1)ooo.TextOut(daptx*frmX-2+posBYx+ posX, dlg.y-
12+posBYy ,      ct_); //y achsen bezeichnung unten
                         if( sw_status)if(sw_yb==1)ooo.TextOut(daptx*frmX-2+posBYx+ posX, dlg.y-
12+posBYy-12 ,   ct_); //


                         ooo.SelectObject(&o1s);
                         ooo.SelectObject(&of2);
                         ooo.SetBkColor( fb_hg);//hintergrundfarbe
                     ooo.SetTextColor(Ax_fb);


                                      float dapty = egy  *(mv2y/(e_y/sc)) ; //dialog-achsen-
koordinatenpunkt y
                             if(sw_mkoord_A)dapty = ( dlg.y -(((((mAy))) +sc0y)/(max_y+sc0y))*
dy_s)-dy_sy; //dialog-Achsen-koordinatenpunkt y manuell
                                                              //~~~
                                if(!sw_mkoord_A)Ay_m= (((( egy*((1-mv2y)/(e_y/sc)))-
dy_sy)/dy_s)*(max_y+sc0y))-sc0y; //global für koordinatenübergabe an
koordinateneinstellungsdialog
                  //if( sw_mkoord_A)Ay_m=mAy;
                                if(!sw_mkoord_A)     mAy=Ay_m;

                         if(sw_xA==1)
                         ooo.MoveTo( 0,     dapty*frmY+ posY) ;   //x achse
                         if(sw_xA==1)
```

```
                        ooo.LineTo( dlg.x, dapty*frmY+ posY);    //


                                                                               ct_="x";
                                                                                              if(sw_inv==1)
ct_="y";     //f-1(x)
                                                                                              if(sw_mod_==2)
ct_="z(x)";  //rxy

if(sw_mod_==3)if(p_p_sw==0)if(sw_pq==0) ct_="p(x)";  //Fp

if(sw_mod_==3)if(p_p_sw==0)if(sw_pq==1) ct_="p(y)";  //

if(sw_mod_==3)if(p_p_sw==1)if(sw_pq==0) ct_="pa1(x)";//

if(sw_mod_==3)if(p_p_sw==1)if(sw_pq==1) ct_="pa1(y)";//

if(sw_mod_==3)if(p_p_sw==2)if(sw_pq==0) ct_="pa2(x)";//

if(sw_mod_==3)if(p_p_sw==2)if(sw_pq==1) ct_="pa2(y)";//

if(sw_mod_==3)if(sw_inv==1)if(sw_pq==0) ct_="x=q";   //

if(sw_mod_==3)if(sw_inv==1)if(sw_pq==1) ct_="y=q";   //

if(sw_mod_==4)if(sw_inv==1)if(p_e_sw==0)ct_="p";     //Fe

if(sw_mod_==4)if(sw_inv==1)if(p_e_sw==1)ct_="pa1";   //

if(sw_mod_==4)if(sw_inv==1)if(p_e_sw==2)ct_="pa2";   //

if(sw_mod_==4)if(sw_inv==0)             ct_="q";     //

                                                                                              if(sw_xbz)
ct_=xBz;     //manuell
                        if(sw_xA==1)
                        if(sw_xb==1)ooo.TextOut(      0+posBXx , dapty*frmY-6+posBXy+ posY,
ct_); //x achsen bezeichnung links
                        if(sw_xA==1)
                        if(sw_xb>=1)ooo.TextOut(dlg.x-6+posBXx , dapty*frmY-6+posBXy+ posY ,
ct_); //x achsen bezeichnung rechts


        ///////////////////////////////////////////////////////////////////////////////////////
/////////////////

                if(sw_drw)//dynamisches rendern (rerender schalter)
                {
                        int il=0;

                        int x; int y;

                        int yr;  int ys0; int ysi0; int yss0; int yssi0; int ysr0; int ysri0;
                        int yri; int ys1; int ysi1; int yss1; int yssi1; int ysr1; int ysri1;

                        float x_m_min; float y_m_min;
                        float x_m_max; float y_m_max;

                        float yr_m_min; float ys0_m_min;float ysi0_m_min;float yss0_m_min;float
yssi0_m_min;float ysr0_m_min;float ysri0_m_min;
                        float yr_m_max; float ys0_m_max;float ysi0_m_max;float yss0_m_max;float
yssi0_m_max;float ysr0_m_max;float ysri0_m_max;
                        float yri_m_min;float ys1_m_min;float ysi1_m_min;float yss1_m_min;float
yssi1_m_min;float ysr1_m_min;float ysri1_m_min;
                        float yri_m_max;float ys1_m_max;float ysi1_m_max;float yss1_m_max;float
yssi1_m_max;float ysr1_m_max;float ysri1_m_max;



        ///////////////////////////////////////////////////////////////////////////////////////
//////////////////////////
                        // Funktionswertschleife
///////////////////////////////////////////////////////////////////////////////

        ///////////////////////////////////////////////////////////////////////////////////////
//////////////////////////

                        FILE *f_;
```

```
                                f_=fopen(filename,"r"); //über die funktionsmatrixdatei

                      if(filestr)do //filestream rendern
                      {
                                                                            char cx_[20], 
cy_[20];

                              if(sw_inv==0||(sw_inv==1&&sw_mod_==2)) fscanf(f_,"%s%s",&cx_, 
&cy_); //funktionswerte einlesen
                              if(sw_inv==1)          if(sw_mod_!=2)  fscanf(f_,"%s%s",&cy_, 
&cx_); //f-1


                              x_=atof(cx_); x_+=sc0x;
                              y_=atof(cy_); y_+=sc0y;


                                             char ccx_[20];
                                      sprintf(ccx_,"%s",      ftoc(x_-sc0x, ds_x)); 
//formatierte skalenwerte x
                              if(0)sprintf(ccx_,"%g", atof(ftoc(x_-sc0x, ds_x)));//zu 
implementieren...

                                             char ccy_[20];
                                      sprintf(ccy_,"%s",      ftoc(y_-sc0y, ds_y)); 
//formatierte skalenwerte y
                              if(0)sprintf(ccx_,"%g", atof(ftoc(x_-sc0x, ds_x)));//zu 
implementieren...


                              //koordinatenpunkt berechnung

                                             float pr_x = x_/(max_x+sc0x); //xi proportion
                                             float pr_y = y_/(max_y+sc0y); //yi proportion

                               x=           pr_x * dx_s  ; //dialog koordinatenpunkt zu wert x
                               y= dlg.y - ( pr_y * dy_s ); //dialog koordinatenpunkt zu wert y

                               x+=dx_sy;
                               y-=dy_sy;


////////////////////////////////////////////////////////////////////////////////////////////////
/////////////
                               // THETA rxy berechnung
                               //

                               if(sw_mod_==2) //Regressionsgeradenwert y
                               {
                                                                      float pr_yr;
                                                                      float pr_yri;
                                       if(sw_inv==0||rxy_D==1)pr_yr  = ((atof(cx_)* qR )
+sc0y)/(max_y+sc0y); //xir   proportion
                                       if(sw_inv==1||rxy_D==1)pr_yri = ((atof(cy_)* qR )
+sc0x)/(max_x+sc0x); //xir-1 proportion

                                      if(sw_inv==0||rxy_D==1)yr= dlg.y - ( pr_yr * dy_s ); 
//dialog koordinatenpunkt y'x
                                      if(sw_inv==0||rxy_D==1)yr-=dy_sy;

                                      if(sw_inv==1||rxy_D==1)yri= pr_yri * dx_s; 
//dialog koordinatenpunkt x'y
                                      if(sw_inv==1||rxy_D==1)yri+=dx_sy;
                               }

                               if(sw_mod_==2)if(1) //Standardvorhersagefehler sy'x
                               {
                                                                          float pr_ys0;
                                                                          float pr_ys1;
                                                                          float pr_ysi0;
                                                                          float pr_ysi1;
                                       if(sw_inv==0||rxy_D==1) pr_ys0 =  (( (atof(cx_)* qR )-
qS*ci_zr) +sc0y)/(max_y+sc0y); //sy'- proportion
                                       if(sw_inv==0||rxy_D==1) pr_ys1 =  (( (atof(cx_)* qR
)+qS*ci_zr) +sc0y)/(max_y+sc0y); //sy'+ proportion
                                       if(sw_inv==1||rxy_D==1) pr_ysi0 = (( (atof(cy_)* qR )-
qS*ci_zr) +sc0x)/(max_x+sc0x); //sx'- proportion
```

```
                                       if(sw_inv==1||rxy_D==1) pr_ysi1 = (( (atof(cy_)* qR
)+qS*ci_zr) +sc0x)/(max_x+sc0x); //sx'+ proportion

                                      if(sw_inv==0||rxy_D==1)
                                      {
                                              ys0= dlg.y - ( pr_ys0 * dy_s ); //dialog
koordinatenpunkt sy'x-
                                              ys0-=dy_sy;

                                              ys1= dlg.y - ( pr_ys1 * dy_s ); //dialog
koordinatenpunkt sy'x+
                                              ys1-=dy_sy;
                                      }

                                      if(sw_inv==1||rxy_D==1)
                                      {
                                              ysi0=  pr_ysi0 * dx_s ;           //dialog
koordinatenpunkt sx'y-
                                              ysi0+=dx_sy;

                                              ysi1=  pr_ysi1 * dx_s ;           //dialog
koordinatenpunkt sx'y+
                                              ysi1+=dx_sy;
                                      }
                               }

                               if(sw_mod_==2)if(1) //Geschätzter Standardvorhersagefehler
s'y'x
                               {
                                                                          float pr_yss0;
                                                                          float pr_yss1;
                                                                          float pr_yssi0;
                                                                          float pr_yssi1;
                                       if(sw_inv==0||rxy_D==1) pr_yss0  = (( (atof(cx_)* qR )-
qsS*ci_zr) +sc0y)/(max_y+sc0y); //sy'- proportion
                                       if(sw_inv==0||rxy_D==1) pr_yss1  = (( (atof(cx_)* qR
)+qsS*ci_zr) +sc0y)/(max_y+sc0y); //sy'+ proportion
                                       if(sw_inv==1||rxy_D==1) pr_yssi0 = (( (atof(cy_)* qR )-
qsS*ci_zr) +sc0x)/(max_x+sc0x); //sx'- proportion
                                       if(sw_inv==1||rxy_D==1) pr_yssi1 = (( (atof(cy_)* qR
)+qsS*ci_zr) +sc0x)/(max_x+sc0x); //sx'+ proportion

                                      if(sw_inv==0||rxy_D==1)
                                      {
                                              yss0= dlg.y - ( pr_yss0 * dy_s ); //dialog
koordinatenpunkt s'y'x-
                                              yss0-=dy_sy;

                                              yss1= dlg.y - ( pr_yss1 * dy_s ); //dialog
koordinatenpunkt s'y'x+
                                              yss1-=dy_sy;
                                      }

                                      if(sw_inv==1||rxy_D==1)
                                      {
                                              yssi0=  pr_yssi0 * dx_s ;           //dialog
koordinatenpunkt s'x'y-
                                              yssi0+= dx_sy;

                                              yssi1=  pr_yssi1 * dx_s ;           //dialog
koordinatenpunkt s'x'y+
                                              yssi1+= dx_sy;
                                      }
                               }

                               if(sw_mod_==2)if(1) //Geschätzter Standardfehler der Regression
s'r
                               {
                                                                      float pr_ysr0;
                                                                      float pr_ysr1;
                                                                      float pr_ysri0;
                                                                      float pr_ysri1;
                                      if(sw_inv==0||rxy_D==1) pr_ysr0 = ((atof(cx_)* (qR-
qsR*ci_tr )) +sc0y)/(max_y+sc0y); //xir'   proportion
                                      if(sw_inv==0||rxy_D==1) pr_ysr1 = ((atof(cx_)*
(qR+qsR*ci_tr )) +sc0y)/(max_y+sc0y); //xir'   proportion
                                      if(sw_inv==1||rxy_D==1) pr_ysri0 = ((atof(cy_)* (qR-
qsR*ci_tr )) +sc0x)/(max_x+sc0x); //xir'-1 proportion
```

```
                                      if(sw_inv==1||rxy_D==1) pr_ysri1 = ((atof(cy_)*
(qR+qsR*ci_tr )) +sc0x)/(max_x+sc0x); //xir'-1 proportion

                                      if(sw_inv==0||rxy_D==1)
                                      {
                                             ysr0= dlg.y - ( pr_ysr0 * dy_s ); //dialog
koordinatenpunkt y'x r'
                                             ysr0-=dy_sy;

                                             ysr1= dlg.y - ( pr_ysr1 * dy_s );
                                             ysr1-=dy_sy;
                                      }

                                      if(sw_inv==1||rxy_D==1)
                                      {
                                             ysri0= pr_ysri0 * dx_s;             //dialog
koordinatenpunkt x'y r'
                                             ysri0+=dx_sy;

                                             ysri1= pr_ysri1 * dx_s;
                                             ysri1+=dx_sy;
                                      }
                               }


//////////////////////////////////////////////////////////////////////////////////////////////
/////////////
                               // Funktionszeichnung
                               //

                              ooo.SelectObject(&b1);

                              CRect xy_(x*frmX-linB_FnP+ posX,  y*frmY-linB_FnP+ posY,
                                           x*frmX+linB_FnP+ posX,  y*frmY+linB_FnP+ posY);//xy
punkt

                              POINT xy_1;
                                       xy_1.x=x*frmX+linB_FnP+ posX;
                                       xy_1.y=y*frmY+linB_FnP+ posY;

                               if(swli==1)// erster wert
                               {
                                     ooo.MoveTo (x*frmX+ posX,  y*frmY+ posY );

                                     if(sw_xy==1) // pixel setzen x xy Punkt
                                     {
                                            ooo.SelectObject(&o2s1);

                                            if(mod_FnP==3)//kreuz
                                            {
                                                   ooo.MoveTo (xy_.left,xy_.top);
                                                   ooo.LineTo (xy_.right,xy_.bottom);
                                                   ooo.MoveTo (xy_.right,xy_.top);
                                                   ooo.LineTo (xy_.left,xy_.bottom);
                                            }

       if(mod_FnP==2)ooo.FrameRect(xy_,&b2);//ooo.FillSolidRect(xy_,fb_P)//eckig
                                            if(mod_FnP==1)ooo.RoundRect(xy_, xy_1); // rund
                                     }

                                     if(sw_x==1)//funktionswert x ausgeben
                                     {
                                            ooo.SelectObject(&ofx);
                                            ooo.SetTextColor(fn_x_fb);
                                            ooo.SetBkMode(TRANSPARENT);
                                            ooo.TextOut(x+ posX,  dlg.y-12+ posY,
ccx_);
                                     }

                                     if(sw_y==1)//funktionswert y ausgeben
                                     {
                                            ooo.SelectObject(&ofy);
                                            ooo.SetTextColor(fn_y_fb);
                                            ooo.SetBkMode(TRANSPARENT);
                                            ooo.TextOut(6+ posX,      y-12+ posY,
ccy_);
                                     }
```

```
                                      swli=0;// erster wert schalter

                                      //minima maxima
                                      x_m_min=x; y_m_min=y;
                                      x_m_max=x; y_m_max=y;

                                      yr_m_min=yr;   ys0_m_min=ys0; ysi0_m_min=ysi0;
yss0_m_min=yss0; yssi0_m_min=yssi0; ysr0_m_min=ysr0;ysri0_m_min=ysri0;
                                      yr_m_max=yr;   ys0_m_max=ys0; ysi0_m_max=ysi0;
yss0_m_max=yss0; yssi0_m_max=yssi0; ysr0_m_max=ysr0;ysri0_m_max=ysri0;
                                      yri_m_min=yri; ys1_m_min=ys1; ysi1_m_min=ysi1;
yss1_m_min=yss1; yssi1_m_min=yssi1; ysr1_m_min=ysr1;ysri1_m_min=ysri1;
                                      yri_m_max=yri; ys1_m_max=ys1; ysi1_m_max=ysi1;
yss1_m_max=yss1; yssi1_m_max=yssi1; ysr1_m_max=ysr1;ysri1_m_max=ysri1;
                               }

                               if(sw_FK==1)
                               if(swli==0) // folgende werte ----- Funktionskurve
                               {
                                     ooo.SelectObject(&o2s);
                                     if(sw_mod_==4)//linienunterbrechung bei F(e)
                                     {
                                           il++;
                                        if(il==(n_/2)+1){ooo.MoveTo (x*frmX+ posX,  y*frmY+
posY );}

                                        else {ooo.LineTo (x*frmX+ posX,  y*frmY+ posY );}
                                     }
                                     else
                                     {ooo.LineTo (x*frmX+ posX,  y*frmY+ posY );}
                               }

                               if(sw_xy==1)// pixel setzen x xy Punkte
                               {
                                     ooo.SelectObject(&o2s1);

                                     if(mod_FnP==3)//kreuz
                                     {
                                            ooo.MoveTo (xy_.left,xy_.top);
                                            ooo.LineTo (xy_.right,xy_.bottom);
                                            ooo.MoveTo (xy_.right,xy_.top);
                                            ooo.LineTo (xy_.left,xy_.bottom);
                                            ooo.MoveTo (x*frmX+ posX,  y*frmY+ posY );
                                     }

       if(mod_FnP==2)ooo.FrameRect(xy_,&b2);//ooo.FillSolidRect(xy_,fb_P)//eckig
                                     if(mod_FnP==1)ooo.RoundRect(xy_, xy_1); // rund
                               }

                               if(sw_x==1)//achsen beschriftung funktions werte
                               {
                                      ooo.SelectObject(&ofx);
                                      ooo.SetTextColor(fn_x_fb);
                                      ooo.SetBkMode(TRANSPARENT);

                                      ooo.TextOut(x*frmX +posFXx+ posX,  dlg.y*frmY +posFXy+
posY,    ccx_);
                               }

                               if(sw_y==1)
                               {
                                      ooo.SelectObject(&ofy);
                                      ooo.SetTextColor(fn_y_fb);
                                      ooo.SetBkMode(TRANSPARENT);

                                      ooo.TextOut(0*frmX +posFYx+ posX,  y*frmY +posFYy+ posY,
ccy_);
                               }

                               if(sw_xm==1) //achsen beschriftung funktions minmax
                               {
                                      ooo.SelectObject(&ofx);
                                      ooo.SetTextColor(fn_x_fb);
                                      ooo.SetBkMode(TRANSPARENT);

                                      if(x_==min_x+sc0x) ooo.TextOut(x*frmX +posFXx+ posX,
dlg.y*frmY +posFXy+ posY,    ccx_);
```

```
                                      if(x_==max_x+sc0x) ooo.TextOut(x*frmX +posFXx+ posX,
dlg.y*frmY +posFXy+ posY,    ccx_);
                               }

                               if(sw_ym==1)
                               {
                                      ooo.SelectObject(&ofy);
                                      ooo.SetTextColor(fn_y_fb);
                                      ooo.SetBkMode(TRANSPARENT);

                                      if(y_==min_y+sc0y)ooo.TextOut(0*frmX +posFYx+ posX,
y*frmY +posFYy+ posY,          ccy_);
                                      if(y_==max_y+sc0y)ooo.TextOut(0*frmX +posFYx+ posX,
y*frmY +posFYy+ posY,          ccy_);
                               }

                               //achsenskalen- und regressionsmarkierungs variablen

                               if(x> x_m_max) x_m_max=x; if(y> y_m_max) y_m_max=y;
                               if(x< x_m_min) x_m_min=x; if(y< y_m_min) y_m_min=y;

                               if(yr>  yr_m_max)  yr_m_max=yr;   if(ys0> ys0_m_max)
ys0_m_max=ys0; if(ysi0> ysi0_m_max) ysi0_m_max=ysi0;
                               if(yr<  yr_m_min)  yr_m_min=yr;   if(ys0< ys0_m_min)
ys0_m_min=ys0; if(ysi0< ysi0_m_min) ysi0_m_min=ysi0;
                               if(yri> yri_m_max) yri_m_max=yri; if(ys1> ys1_m_max)
ys1_m_max=ys1; if(ysi1> ysi1_m_max) ysi1_m_max=ysi1;
                               if(yri< yri_m_min) yri_m_min=yri; if(ys1< ys1_m_min)
ys1_m_min=ys1; if(ysi1< ysi1_m_min) ysi1_m_min=ysi1;

                               if(ysr0> ysr0_m_max) ysr0_m_max=ysr0;  if(ysr1> ysr1_m_max)
ysr1_m_max=ysr1;  if(yss0> yss0_m_max) yss0_m_max=yss0;
                               if(ysr0< ysr0_m_min) ysr0_m_min=ysr0;  if(ysr1< ysr1_m_min)
ysr1_m_min=ysr1;  if(yss0< yss0_m_min) yss0_m_min=yss0;
                               if(ysri0>ysri0_m_max)ysri0_m_max=ysri0;if(ysri1>ysri1_m_max)
ysri1_m_max=ysri1;if(yss1> yss1_m_max) yss1_m_max=yss1;
                               if(ysri0<ysri0_m_min)ysri0_m_min=ysri0;if(ysri1<ysri1_m_min)
ysri1_m_min=ysri1;if(yss1< yss1_m_min) yss1_m_min=yss1;

                               if(yssi0>yssi0_m_max) yssi0_m_max=yssi0;
                               if(yssi0<yssi0_m_min) yssi0_m_min=yssi0;
                               if(yssi1>yssi1_m_max) yssi1_m_max=yssi1;
                               if(yssi1<yssi1_m_min) yssi1_m_min=yssi1;

                      }while ( feof (f_) == 0); //
                                     fclose(f_);

                      int ni_=0;// über Funktionsvektoren

                      if(!filestr)do  //nicht filestream rendern
                      {

float fx_,            fy_;
                                                                      fx_=FVx_[ni_];
fy_=FVy_[ni_];
                                                                   x_=fx_+sc0x;
                                                                   y_=
fy_+sc0y;


                      /// einfügen
////////////////////////////////////////////////////////////////////////////////////
                      //  |        // atof(cx_)=fx_, atof(cy_)=fy_
                      //  v

                                             char ccx_[20];
                                      sprintf(ccx_,"%s",      ftoc(x_-sc0x, ds_x));
//formatierte skalenwerte x
                              if(0)sprintf(ccx_,"%g", atof(ftoc(x_-sc0x, ds_x)));//zu
implementieren...

                                             char ccy_[20];
                                      sprintf(ccy_,"%s",      ftoc(y_-sc0y, ds_y));
//formatierte skalenwerte y
                              if(0)sprintf(ccx_,"%g", atof(ftoc(x_-sc0x, ds_x)));//zu
implementieren...
```

```
                              //koordinatenpunkt berechnung

                                              float pr_x = x_/(max_x+sc0x); //xi proportion
                                              float pr_y = y_/(max_y+sc0y); //yi proportion

                                x=           pr_x * dx_s  ; //dialog koordinatenpunkt zu wert x
                                y= dlg.y - ( pr_y * dy_s ); //dialog koordinatenpunkt zu wert y

                                x+=dx_sy;
                                y-=dy_sy;


/////////////////////////////////////////////////////////////////////////////////////////////////
/////////////
                                // THETA rxy berechnung
                                //

                                if(sw_mod_==2) //Regressionsgeradenwert y
                                {
                                                                        float pr_yr;
                                                                        float pr_yri;
                                        if(sw_inv==0||rxy_D==1)pr_yr  = ((fx_* qR )
+sc0y)/(max_y+sc0y); //xir   proportion
                                        if(sw_inv==1||rxy_D==1)pr_yri = ((fy_* qR )
+sc0x)/(max_x+sc0x); //xir-1 proportion

                                       if(sw_inv==0||rxy_D==1)yr= dlg.y - ( pr_yr * dy_s );
//dialog koordinatenpunkt y'x
                                       if(sw_inv==0||rxy_D==1)yr-=dy_sy;

                                       if(sw_inv==1||rxy_D==1)yri= pr_yri * dx_s;
//dialog koordinatenpunkt x'y
                                       if(sw_inv==1||rxy_D==1)yri+=dx_sy;
                                }

                                if(sw_mod_==2)if(1) //Standardvorhersagefehler sy'x
                                {
                                                                            float pr_ys0;
                                                                            float pr_ys1;
                                                                            float pr_ysi0;
                                                                            float pr_ysi1;
                                        if(sw_inv==0||rxy_D==1) pr_ys0 =  (( (fx_* qR )-
qS*ci_zr) +sc0y)/(max_y+sc0y); //sy'- proportion
                                        if(sw_inv==0||rxy_D==1) pr_ys1 =  (( (fx_* qR
)+qS*ci_zr) +sc0y)/(max_y+sc0y); //sy'+ proportion
                                        if(sw_inv==1||rxy_D==1) pr_ysi0 = (( (fy_* qR )-
qS*ci_zr) +sc0x)/(max_x+sc0x); //sx'- proportion
                                        if(sw_inv==1||rxy_D==1) pr_ysi1 = (( (fy_* qR
)+qS*ci_zr) +sc0x)/(max_x+sc0x); //sx'+ proportion

                                       if(sw_inv==0||rxy_D==1)
                                       {
                                               ys0= dlg.y - ( pr_ys0 * dy_s ); //dialog
koordinatenpunkt sy'x-
                                               ys0-=dy_sy;

                                               ys1= dlg.y - ( pr_ys1 * dy_s ); //dialog
koordinatenpunkt sy'x+
                                               ys1-=dy_sy;
                                       }

                                       if(sw_inv==1||rxy_D==1)
                                       {
                                               ysi0=  pr_ysi0 * dx_s ;           //dialog
koordinatenpunkt sx'y-
                                               ysi0+=dx_sy;

                                               ysi1=  pr_ysi1 * dx_s ;           //dialog
koordinatenpunkt sx'y+
                                               ysi1+=dx_sy;
                                       }
                                }

                                if(sw_mod_==2)if(1) //Geschätzter Standardvorhersagefehler
s'y'x
                                {
                                                                            float pr_yss0;
                                                                            float pr_yss1;
                                                                            float pr_yssi0;
                                                                            float pr_yssi1;
                                        if(sw_inv==0||rxy_D==1) pr_yss0  = (( (fx_* qR )-
qsS*ci_zr) +sc0y)/(max_y+sc0y); //sy'- proportion
                                        if(sw_inv==0||rxy_D==1) pr_yss1  = (( (fx_* qR
)+qsS*ci_zr) +sc0y)/(max_y+sc0y); //sy'+ proportion
                                        if(sw_inv==1||rxy_D==1) pr_yssi0 = (( (fy_* qR )-
qsS*ci_zr) +sc0x)/(max_x+sc0x); //sx'- proportion
                                        if(sw_inv==1||rxy_D==1) pr_yssi1 = (( (fy_* qR
)+qsS*ci_zr) +sc0x)/(max_x+sc0x); //sx'+ proportion

                                       if(sw_inv==0||rxy_D==1)
                                       {
                                              yss0= dlg.y - ( pr_yss0 * dy_s ); //dialog
koordinatenpunkt s'y'x-
                                              yss0-=dy_sy;

                                              yss1= dlg.y - ( pr_yss1 * dy_s ); //dialog
koordinatenpunkt s'y'x+
                                              yss1-=dy_sy;
                                       }

                                       if(sw_inv==1||rxy_D==1)
                                       {
                                              yssi0=  pr_yssi0 * dx_s ;           //dialog
koordinatenpunkt s'x'y-
                                              yssi0+= dx_sy;

                                              yssi1=  pr_yssi1 * dx_s ;           //dialog
koordinatenpunkt s'x'y+
                                              yssi1+= dx_sy;
                                       }
                               }

                               if(sw_mod_==2)if(1) //Geschätzter Standardfehler der Regression
s'r
                               {
                                                                       float pr_ysr0;
                                                                       float pr_ysr1;
                                                                       float pr_ysri0;
                                                                       float pr_ysri1;
                                       if(sw_inv==0||rxy_D==1) pr_ysr0 = ((fx_* (qR-qsR*ci_tr
)) +sc0y)/(max_y+sc0y); //xir'   proportion
                                       if(sw_inv==0||rxy_D==1) pr_ysr1 = ((fx_* (qR+qsR*ci_tr
)) +sc0y)/(max_y+sc0y); //xir'   proportion
                                       if(sw_inv==1||rxy_D==1) pr_ysri0 = ((fy_* (qR-qsR*ci_tr
)) +sc0x)/(max_x+sc0x); //xir'-1 proportion
                                       if(sw_inv==1||rxy_D==1) pr_ysri1 = ((fy_* (qR+qsR*ci_tr
)) +sc0x)/(max_x+sc0x); //xir'-1 proportion

                                       if(sw_inv==0||rxy_D==1)
                                       {
                                              ysr0= dlg.y - ( pr_ysr0 * dy_s ); //dialog
koordinatenpunkt y'x r'
                                              ysr0-=dy_sy;

                                              ysr1= dlg.y - ( pr_ysr1 * dy_s );
                                              ysr1-=dy_sy;
                                       }

                                       if(sw_inv==1||rxy_D==1)
                                       {
                                              ysri0= pr_ysri0 * dx_s;             //dialog
koordinatenpunkt x'y r'
                                              ysri0+=dx_sy;

                                              ysri1= pr_ysri1 * dx_s;
                                              ysri1+=dx_sy;
                                       }
                               }

////////////////////////////////////////////////////////////////////////////////////////////////
/////////////
                               // Funktionszeichnung
                               //

                              ooo.SelectObject(&b1);
                              CRect xy_(x*frmX-linB_FnP+ posX,y*frmY-linB_FnP+ posY,
                                            x*frmX+linB_FnP+ posX,y*frmY+linB_FnP+ posY);//xy
punkt

                              POINT xy_1;
                                        xy_1.x=x*frmX+linB_FnP+ posX;
                                        xy_1.y=y*frmY+linB_FnP+ posY;

                               if(swli==1)
                               {
                                      ooo.MoveTo (x*frmX+ posX,  y*frmY+ posY ); // erster wert

                                      if(sw_xy==1) // pixel setzen x xy Punkt
                                      {
                                             ooo.SelectObject(&o2s1);

                                             if(mod_FnP==3)//kreuz
                                             {
                                                    ooo.MoveTo (xy_.left,xy_.top);
                                                    ooo.LineTo (xy_.right,xy_.bottom);
                                                    ooo.MoveTo (xy_.right,xy_.top);
                                                    ooo.LineTo (xy_.left,xy_.bottom);
                                             }

        if(mod_FnP==2)ooo.FrameRect(xy_,&b2);//ooo.FillSolidRect(xy_,fb_P)//eckig
                                             if(mod_FnP==1)ooo.RoundRect(xy_, xy_1); // rund
                                      }

                                      if(sw_x==1)//funktionswert x ausgeben
                                      {
                                             ooo.SelectObject(&ofx);
                                             ooo.SetTextColor(fn_x_fb);
                                             ooo.SetBkMode(TRANSPARENT);
                                             ooo.TextOut(x*frmX+ posX,  dlg.y*frmY-12+ posY,
ccx_);
                                      }

                                      if(sw_y==1)//funktionswert y ausgeben
                                      {
                                             ooo.SelectObject(&ofy);
                                             ooo.SetTextColor(fn_y_fb);
                                             ooo.SetBkMode(TRANSPARENT);
                                             ooo.TextOut(6*frmX+ posX,      y*frmY-12+ posY,
ccy_);
                                      }

                                       swli=0;//

                                       //minima maxima
                                       x_m_min=x; y_m_min=y;
                                       x_m_max=x; y_m_max=y;

                                       yr_m_min=yr;   ys0_m_min=ys0; ysi0_m_min=ysi0;
yss0_m_min=yss0; yssi0_m_min=yssi0; ysr0_m_min=ysr0;ysri0_m_min=ysri0;
                                       yr_m_max=yr;   ys0_m_max=ys0; ysi0_m_max=ysi0;
yss0_m_max=yss0; yssi0_m_max=yssi0; ysr0_m_max=ysr0;ysri0_m_max=ysri0;
                                       yri_m_min=yri; ys1_m_min=ys1; ysi1_m_min=ysi1;
yss1_m_min=yss1; yssi1_m_min=yssi1; ysr1_m_min=ysr1;ysri1_m_min=ysri1;
                                       yri_m_max=yri; ys1_m_max=ys1; ysi1_m_max=ysi1;
yss1_m_max=yss1; yssi1_m_max=yssi1; ysr1_m_max=ysr1;ysri1_m_max=ysri1;
                               }

                               if(sw_FK==1)
                               if(swli==0) // folgende werte ----- Funktionskurve
                               {
                                      ooo.SelectObject(&o2s);
                                      if(sw_mod_==4)//linienunterbrechung bei F(e)
                                      {
                                            il++;
                                         if(il==(n_/2)+1){ooo.MoveTo (x*frmX+ posX,  y*frmY+
posY );}

                                         else {ooo.LineTo (x*frmX+ posX,  y*frmY+ posY );}
                                      }
                                      else
                                      {ooo.LineTo (x*frmX+ posX,  y*frmY+ posY );}
                               }
                               if(sw_xy==1)// pixel setzen x xy Punkte
                               {
                                      ooo.SelectObject(&o2s1);

                                      if(mod_FnP==3)//kreuz
                                      {
                                             ooo.MoveTo (xy_.left,xy_.top);
                                             ooo.LineTo (xy_.right,xy_.bottom);
                                             ooo.MoveTo (xy_.right,xy_.top);
                                             ooo.LineTo (xy_.left,xy_.bottom);
                                             ooo.MoveTo (x+ posX,  y );
                                      }

        if(mod_FnP==2)ooo.FrameRect(xy_,&b2);//ooo.FillSolidRect(xy_,fb_P)//eckig
                                      if(mod_FnP==1)ooo.RoundRect(xy_, xy_1); // rund
                               }



                               if(sw_x==1)//achsen beschriftung funktions werte
                               {
                                       ooo.SelectObject(&ofx);
                                       ooo.SetTextColor(fn_x_fb);
                                       ooo.SetBkMode(TRANSPARENT);

                                       ooo.TextOut(x*frmX +posFXx+ posX,  dlg.y*frmY +posFXy+
posY,    ccx_);
                               }

                               if(sw_y==1)
                               {
                                       ooo.SelectObject(&ofy);
                                       ooo.SetTextColor(fn_y_fb);
                                       ooo.SetBkMode(TRANSPARENT);

                                       ooo.TextOut(0*frmX +posFYx+ posX,  y*frmY +posFYy+ posY,
ccy_);
                               }

                               if(sw_xm==1) //achsen beschriftung funktions minmax
                               {
                                       ooo.SelectObject(&ofx);
                                       ooo.SetTextColor(fn_x_fb);
                                       ooo.SetBkMode(TRANSPARENT);

                                       if(x_==min_x+sc0x) ooo.TextOut(x*frmX +posFXx+ posX,
dlg.y*frmY +posFXy+ posY,    ccx_);
                                       if(x_==max_x+sc0x) ooo.TextOut(x*frmX +posFXx+ posX,
dlg.y*frmY +posFXy+ posY,    ccx_);
                               }

                               if(sw_ym==1)
                               {
                                       ooo.SelectObject(&ofy);
                                       ooo.SetTextColor(fn_y_fb);
                                       ooo.SetBkMode(TRANSPARENT);

                                       if(y_==min_y+sc0y)ooo.TextOut(0*frmX +posFYx+ posX,
y*frmY +posFYy+ posY,          ccy_);
                                       if(y_==max_y+sc0y)ooo.TextOut(0*frmX +posFYx+ posX,
y*frmY +posFYy+ posY,          ccy_);
                               }

                               //achsenskalen- und regressionsmarkierungs variablen

                               if(x> x_m_max) x_m_max=x; if(y> y_m_max) y_m_max=y;
                               if(x< x_m_min) x_m_min=x; if(y< y_m_min) y_m_min=y;

                               if(yr>  yr_m_max)  yr_m_max=yr;   if(ys0> ys0_m_max)
ys0_m_max=ys0; if(ysi0> ysi0_m_max) ysi0_m_max=ysi0;
                               if(yr<  yr_m_min)  yr_m_min=yr;   if(ys0< ys0_m_min)
ys0_m_min=ys0; if(ysi0< ysi0_m_min) ysi0_m_min=ysi0;
                               if(yri> yri_m_max) yri_m_max=yri; if(ys1> ys1_m_max)
ys1_m_max=ys1; if(ysi1> ysi1_m_max) ysi1_m_max=ysi1;
                               if(yri< yri_m_min) yri_m_min=yri; if(ys1< ys1_m_min)
ys1_m_min=ys1; if(ysi1< ysi1_m_min) ysi1_m_min=ysi1;
```

```
                               if(ysr0> ysr0_m_max) ysr0_m_max=ysr0;  if(ysr1> ysr1_m_max)
ysr1_m_max=ysr1;  if(yss0> yss0_m_max) yss0_m_max=yss0;
                               if(ysr0< ysr0_m_min) ysr0_m_min=ysr0;  if(ysr1< ysr1_m_min)
ysr1_m_min=ysr1;  if(yss0< yss0_m_min) yss0_m_min=yss0;
                               if(ysri0>ysri0_m_max)ysri0_m_max=ysri0;if(ysri1>ysri1_m_max)
ysri1_m_max=ysri1;if(yss1> yss1_m_max) yss1_m_max=yss1;
                               if(ysri0<ysri0_m_min)ysri0_m_min=ysri0;if(ysri1<ysri1_m_min)
ysri1_m_min=ysri1;if(yss1< yss1_m_min) yss1_m_min=yss1;

                               if(yssi0>yssi0_m_max) yssi0_m_max=yssi0;
                               if(yssi0<yssi0_m_min) yssi0_m_min=yssi0;
                               if(yssi1>yssi1_m_max) yssi1_m_max=yssi1;
                               if(yssi1<yssi1_m_min) yssi1_m_min=yssi1;

                      //  ^
                      //  |
                      /// einfügen
/////////////////////////////////////////////////////////////////////////////////////////////

                                 ni_++;
                      }while(ni_<n_); //Funktionsvektoren


                      swli=1; ///


       /////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////
                      // Funktionswertschleife ende
////////////////////////////////////////////////////////////////////////////////////

       /////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////



       /////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////
                      // THETA rxy  Achsen zeichnen
                      //

                      if(sw_mod_==2) // Regressionsgerade
                      {
                             ooo.SelectObject(&of1); //font y-achse

                                                                   CString cr_;
                             if(sw_inv==0||rxy_D==1)if(rxy_==1)
                             {
                                    ooo.SetTextColor(fb_r); //textfarbe
                                    ooo.SelectObject(&or1);                cr_="r(xy)";
                                                                    if(r_q==1)cr_+="= ";

if(r_q==1)cr_+=ftoc(qR,3);
                                    if(qR>=0)
                                    {
                                                  ooo.MoveTo (x_m_min*frmX+ posX,
yr_m_max*frmY+ posY );
                                                  ooo.LineTo (x_m_max*frmX+ posX,
yr_m_min*frmY+ posY );
                                           if(r_q)ooo.TextOut(x_m_max*frmX+3+
posX,yr_m_min*frmY-5+ posY,  cr_);//regressionswert rxy
                                    }

                                    if(qR<0)
                                    {
                                                  ooo.MoveTo (x_m_min*frmX+ posX,
yr_m_min*frmY+ posY );
                                                  ooo.LineTo (x_m_max*frmX+ posX,
yr_m_max*frmY+ posY );
                                           if(r_q)ooo.TextOut(x_m_max*frmX+3+
posX,yr_m_max*frmY-5+ posY,  cr_);//regressionswert rxy
                                    }
                             }

                             if(sw_inv==1||rxy_D==1)if(ryx_==1)
                             {
                                    ooo.SetTextColor(fb_ri); //textfarbe
                                    ooo.SelectObject(&or1i);                cr_="r(yx)";
```

```
                                                                  if(ri_q==1)cr_+="= ";
                                                                  if(ri_q==1)cr_+=ftoc(qR,3);
                                     if(qR>=0)
                                     {
                                                     ooo.MoveTo (yri_m_max*frmX+ posX, 
y_m_min*frmY+ posY);
                                                     ooo.LineTo (yri_m_min*frmX+ posX, 
y_m_max*frmY+ posY);
                                             if(ri_q)ooo.TextOut(yri_m_min*frmX+3+
posX,y_m_max*frmY-5+ posY,  cr_);//regressionswert ryx
                                     }

                                     if(qR<0)
                                     {
                                                     ooo.MoveTo (yri_m_max*frmX+ posX, 
y_m_max*frmY+ posY);
                                                     ooo.LineTo (yri_m_min*frmX+ posX, 
y_m_min*frmY+ posY);
                                             if(ri_q)ooo.TextOut(yri_m_min*frmX+3+
posX,y_m_min*frmY-5+ posY,  cr_);//regressionswert ryx
                                     }
                             }
                      }

                      if(sw_mod_==2) // Standardvorhersagefehler
                      {
                              ooo.SelectObject(&of1); //font y-achse

                              if(sw_inv==0||rxy_D==1)if(syx_==1)
                              {
                                     ooo.SetTextColor(fb_syx); //textfarbe
                                     ooo.SelectObject(&or4);                CString 
cr_="□·sy'x";
                                                                        if(syx_q==1)cr_+="= ";

          if(syx_q==1)cr_+=ftoc(qS*ci_zr,3);
                                     if(qR>=0)
                                     {
                                                      ooo.MoveTo (x_m_min*frmX+ posX, 
ys0_m_max*frmY+ posY );
                                                      ooo.LineTo (x_m_max*frmX+ posX, 
ys0_m_min*frmY+ posY );
                                             if(syx_q)ooo.TextOut(x_m_max*frmX+3+
posX,ys0_m_min*frmY-5+ posY,  cr_);// sy'x wert
                                                      ooo.MoveTo (x_m_min*frmX+ posX, 
ys1_m_max*frmY+ posY );
                                                      ooo.LineTo (x_m_max*frmX+ posX, 
ys1_m_min*frmY+ posY );
                                             if(syx_q)ooo.TextOut(x_m_max*frmX+3+
posX,ys1_m_min*frmY-5+ posY,  cr_);// sy'x wert
                                     }

                                     if(qR<0)
                                     {
                                                      ooo.MoveTo (x_m_min*frmX+ posX, 
ys0_m_min*frmY+ posY );
                                                      ooo.LineTo (x_m_max*frmX+ posX, 
ys0_m_max*frmY+ posY );
                                             if(syx_q)ooo.TextOut(x_m_max*frmX+3+
posX,ys0_m_max*frmY-5+ posY,  cr_);// sy'x wert
                                                      ooo.MoveTo (x_m_min*frmX+ posX, 
ys1_m_min*frmY+ posY );
                                                      ooo.LineTo (x_m_max*frmX+ posX, 
ys1_m_max*frmY+ posY );
                                             if(syx_q)ooo.TextOut(x_m_max*frmX+3+
posX,ys1_m_max*frmY-5+ posY,  cr_);// sy'x wert

                                     }
                              }
                              if(sw_inv==1||rxy_D==1)if(sxy_==1)
                              {
                                     ooo.SetTextColor(fb_sxy); //textfarbe
                                     ooo.SelectObject(&or4i);               CString 
cr_="□·sx'(y)";
                                                                        if(sxy_q==1)cr_+="= ";

          if(sxy_q==1)cr_+=ftoc(qS*ci_zr,3);
                                     if(qR>=0)
                                     {
                                                      ooo.MoveTo (ysi0_m_max*frmX+
posX,y_m_min*frmY+ posY );
                                                      ooo.LineTo (ysi0_m_min*frmX+
posX,y_m_max*frmY+ posY );
                                             if(sxy_q)ooo.TextOut(ysi0_m_min*frmX+3+
posX,y_m_max*frmY-5+ posY,  cr_);// sx'y wert
                                                      ooo.MoveTo (ysi1_m_max*frmX+
posX,y_m_min*frmY+ posY );
                                                      ooo.LineTo (ysi1_m_min*frmX+
posX,y_m_max*frmY+ posY );
                                             if(sxy_q)ooo.TextOut(ysi1_m_min*frmX+3+
posX,y_m_max*frmY-5+ posY,  cr_);// sx'y wert

                                     }
                                     if(qR<0)
                                     {
                                                      ooo.MoveTo (ysi0_m_max*frmX+
posX,y_m_max*frmY+ posY );
                                                      ooo.LineTo (ysi0_m_min*frmX+
posX,y_m_min*frmY+ posY );
                                             if(sxy_q)ooo.TextOut(ysi0_m_min*frmX+3+
posX,y_m_min*frmY-5+ posY,  cr_);// sx'y wert
                                                             ooo.MoveTo (ysi1_m_max*frmX+
posX,y_m_max*frmY+ posY );
                                                      ooo.LineTo (ysi1_m_min*frmX+
posX,y_m_min*frmY+ posY );
                                             if(sxy_q)ooo.TextOut(ysi1_m_min*frmX+3+
posX,y_m_min*frmY-5+ posY,  cr_);// sx'y wert
                                     }
                              }
                      }

                      if(sw_mod_==2) // Geschätzter Standardvorhersagefehler
                      {
                              ooo.SelectObject(&of1); //font y-achse

                              if(sw_inv==0||rxy_D==1)if(s1yx_==1)
                              {
                                     ooo.SetTextColor(fb_sgyx); //textfarbe
                                     ooo.SelectObject(&or5);                CString
cr_="□·ôy'x";
                                                                       if(sgyx_q==1)cr_+="= ";

         if(sgyx_q==1)cr_+=ftoc(qsS*ci_zr,3);

                                     if(qR>=0)
                                     {
                                                       ooo.MoveTo (x_m_min*frmX+ posX,
yss0_m_max*frmY+ posY );
                                                       ooo.LineTo (x_m_max*frmX+ posX,
yss0_m_min*frmY+ posY );
                                             if(sgyx_q)ooo.TextOut(x_m_max*frmX+3+
posX,yss0_m_min*frmY-5+ posY,  cr_);// ôy'x wert
                                                       ooo.MoveTo (x_m_min*frmX+ posX,
yss1_m_max*frmY+ posY );
                                                       ooo.LineTo (x_m_max*frmX+ posX,
yss1_m_min*frmY+ posY );
                                             if(sgyx_q)ooo.TextOut(x_m_max*frmX+3+
posX,yss1_m_min*frmY-5+ posY,  cr_);// ôy'x wert
                                     }

                                     if(qR<0)
                                     {
                                                       ooo.MoveTo (x_m_min*frmX+ posX,
yss0_m_min*frmY+ posY );
                                                       ooo.LineTo (x_m_max*frmX+ posX,
yss0_m_max*frmY+ posY );
                                             if(sgyx_q)ooo.TextOut(x_m_max*frmX+3+
posX,yss0_m_max*frmY-5+ posY,  cr_);// ôy'x wert
                                                       ooo.MoveTo (x_m_min*frmX+ posX,
yss1_m_min*frmY+ posY );
                                                       ooo.LineTo (x_m_max*frmX+ posX,
yss1_m_max*frmY+ posY );
                                             if(sgyx_q)ooo.TextOut(x_m_max*frmX+3+
posX,yss1_m_max*frmY-5+ posY,  cr_);// ôy'x wert
                                     }
                              }
                              if(sw_inv==1||rxy_D==1)if(s1xy_==1)
                              {
                                     ooo.SetTextColor(fb_sgxy); //textfarbe
                                     ooo.SelectObject(&or5i);                CString
cr_="□·ôx'y";
                                                                        if(sgxy_q==1)cr_+="=
";

          if(sgxy_q==1)cr_+=ftoc(qsS*ci_zr,3);

                                     if(qR>=0)
                                     {
                                                       ooo.MoveTo (yssi0_m_max*frmX+
posX,y_m_min*frmY+ posY );
                                                       ooo.LineTo (yssi0_m_min*frmX+
posX,y_m_max*frmY+ posY );
                                             if(sgxy_q)ooo.TextOut(yssi0_m_min*frmX+3+
posX,y_m_max*frmY-5+ posY,  cr_);// ôx'y wert
                                                       ooo.MoveTo (yssi1_m_max*frmX+
posX,y_m_min*frmY+ posY );
                                                       ooo.LineTo (yssi1_m_min*frmX+
posX,y_m_max*frmY+ posY );
                                             if(sgxy_q)ooo.TextOut(yssi1_m_min*frmX+3+
posX,y_m_max*frmY-5+ posY,  cr_);// ôx'y wert
                                     }

                                     if(qR<0)
                                     {
                                                       ooo.MoveTo (yssi0_m_max*frmX+
posX,y_m_max*frmY+ posY );
                                                       ooo.LineTo (yssi0_m_min*frmX+
posX,y_m_min*frmY+ posY );
                                             if(sgxy_q)ooo.TextOut(yssi0_m_min*frmX+3+
posX,y_m_min*frmY-5+ posY,  cr_);// ôx'y wert
                                                       ooo.MoveTo (yssi1_m_max*frmX+
posX,y_m_max*frmY+ posY );
                                                       ooo.LineTo (yssi1_m_min*frmX+
posX,y_m_min*frmY+ posY );
                                             if(sgxy_q)ooo.TextOut(yssi1_m_min*frmX+3+
posX,y_m_min*frmY-5+ posY,  cr_);// ôx'y wert
                                     }
                              }
                      }

                      if(sw_mod_==2) // Geschätzter Korrelationsfehler
                      {
                              ooo.SelectObject(&of1); //font y-achse

                              if(sw_inv==0||rxy_D==1)if(srxy_==1)
                              {
                                     ooo.SetTextColor(fb_sr); //textfarbe
                                     ooo.SelectObject(&or2);                CString
cr_="t·ôr(xy)";
                                                                         if(sr_q==1)cr_+="= ";

if(sr_q==1)cr_+=ftoc(qsR*ci_tr,3);
                                     if(qR>=0)
                                     {
                                                     ooo.MoveTo (x_m_min*frmX+ posX,
ysr0_m_max*frmY+ posY );
                                                     ooo.LineTo (x_m_max*frmX+ posX,
ysr0_m_min*frmY+ posY );
                                             if(sr_q)ooo.TextOut(x_m_max*frmX+3+
posX,ysr0_m_min*frmY-5+ posY,  cr_);// ôrxy wert
                                                     ooo.MoveTo (x_m_min*frmX+ posX,
ysr1_m_max*frmY+ posY );
                                                     ooo.LineTo (x_m_max*frmX+ posX,
ysr1_m_min*frmY+ posY );
                                     }

                                     if(qR<0)
                                     {
                                                     ooo.MoveTo (x_m_min*frmX+ posX,
ysr0_m_min*frmY+ posY );
                                                     ooo.LineTo (x_m_max*frmX+ posX,
ysr0_m_max*frmY+ posY );
                                             if(sr_q)ooo.TextOut(x_m_max*frmX+3+
posX,ysr0_m_max*frmY-5+ posY,  cr_);// ôrxy wert
```

```
                                                    ooo.MoveTo (x_m_min*frmX+ posX,
ysr1_m_min*frmY+ posY );
                                                    ooo.LineTo (x_m_max*frmX+ posX,
ysr1_m_max*frmY+ posY );
                                     }
                              }

                              if(sw_inv==1||rxy_D==1)if(sryx_==1)
                              {
                                     ooo.SetTextColor(fb_sri); //textfarbe
                                     ooo.SelectObject(&or2i);              CString
cr_="t·ôr(yx)";
                                                                         if(sri_q==1)cr_+="=
";

if(sri_q==1)cr_+=ftoc(qsR*ci_tr,3);
                                     if(qR>=0)
                                     {
                                                     ooo.MoveTo (ysri0_m_max*frmX+ posX,
y_m_min*frmY+ posY);
                                                     ooo.LineTo (ysri0_m_min*frmX+ posX,
y_m_max*frmY+ posY);
                                            if(sri_q)ooo.TextOut(ysri0_m_min*frmX+3+
posX,y_m_max*frmY-5+ posY,  cr_);// ôryx wert
                                                     ooo.MoveTo (ysri1_m_max*frmX+ posX,
y_m_min*frmY+ posY);
                                                     ooo.LineTo (ysri1_m_min*frmX+ posX,
y_m_max*frmY+ posY);
                                     }
                                     if(qR<0)
                                     {
                                                     ooo.MoveTo (ysri0_m_max*frmX+ posX,
y_m_max*frmY+ posY);
                                                     ooo.LineTo (ysri0_m_min*frmX+ posX,
y_m_min*frmY+ posY);
                                            if(sri_q)ooo.TextOut(ysri0_m_min*frmX+3+
posX,y_m_min*frmY-5+ posY,  cr_);// ôryx wert
                                                     ooo.MoveTo (ysri1_m_max*frmX+ posX,
y_m_max*frmY+ posY);
                                                     ooo.LineTo (ysri1_m_min*frmX+ posX,
y_m_min*frmY+ posY);
                                     }
                              }
                       }

                       if(sw_mod_==2) // Geschätzter Regressionsfehler
                       {
                              float tmp_0=0;
                              float tmp_1=0;

                              if(sw_inv==0||rxy_D==1)if(srx_==1)
                              {
                                     ooo.SelectObject(&of1); //font y-achse
                                     ooo.SetTextColor(fb_sR); //textfarbe
                                     ooo.SelectObject(&or3);

                                     float dx= ((max_x-min_x) /100);
                                     float dx1=0;
                                     float x_m_1=((x_m_max-x_m_min)/(100));

                                     for(float x_m= x_m_min ;x_m<=x_m_max;x_m+=x_m_1 )
                                     {
                                            float pr_ysp0 = ((( (min_x+dx1)*qR )-(qsS
*sqrt(1/n_+pow(min_x+dx1,2)/n_))*ci_tr) +sc0y)/(max_y+sc0y); //s'Y'- proportion
                                            float pr_ysp1 = ((( (min_x+dx1)*qR )+(qsS
*sqrt(1/n_+pow(min_x+dx1,2)/n_))*ci_tr) +sc0y)/(max_y+sc0y); //s'Y'+ proportion

dx1+=dx;
                                                     pr_ysp0= dlg.y - ( pr_ysp0 * dy_s );
//dialog koordinatenpunkt s'Y'x-
                                                     pr_ysp0-=dy_sy;
                                                     pr_ysp1= dlg.y - ( pr_ysp1 * dy_s );
//dialog koordinatenpunkt s'Y'x+
                                                     pr_ysp1-=dy_sy;

                                            if(x_m>x_m_min)
                                            {
                                                    ooo.MoveTo (x_m*frmX-x_m_1+ posX,
tmp_0*frmY+ posY  );
                                                    ooo.LineTo (x_m*frmX + posX     ,
pr_ysp0*frmY+ posY);

                                                    ooo.MoveTo (x_m*frmX-x_m_1+ posX,
tmp_1*frmY+ posY  );
                                                    ooo.LineTo (x_m*frmX  + posX    ,
pr_ysp1*frmY+ posY);
                                             if(sR_q)if(x_m>=x_m_max-x_m_1)
                                                    ooo.TextOut(x_m*frmX+3+ posX,pr_ysp1*frmY-
5+ posY,  "t·ôR(x)");// ôRx
                                             }

tmp_0=pr_ysp0;

tmp_1=pr_ysp1;
                                      }
                              }

                              if(sw_inv==1||rxy_D==1)if(sry_==1)
                              {
                                      ooo.SelectObject(&of1); //font y-achse
                                      ooo.SetTextColor(fb_sRi); //textfarbe
                                      ooo.SelectObject(&or3i);

                                      float dy= ((max_y-min_y) /(100));
                                      float dy1=0;
                                      float y_m_1=((y_m_max-y_m_min)/(100));

                                      for(float y_m= y_m_max ;y_m>=y_m_min;y_m-=y_m_1 )
                                      {
                                             float pr_ysp0 = ((( (min_y+dy1)*qR )-(qsS
*sqrt(1/n_+pow(min_y+dy1,2)/n_))*ci_tr) +sc0x)/(max_x+sc0x); //s'X'- proportion
                                             float pr_ysp1 = ((( (min_y+dy1)*qR )+(qsS
*sqrt(1/n_+pow(min_y+dy1,2)/n_))*ci_tr) +sc0x)/(max_x+sc0x); //s'X'+ proportion

dy1+=dy;

                                                      pr_ysp0= pr_ysp0 * dx_s ; //dialog
koordinatenpunkt s'X'y-
                                                      pr_ysp0+=dx_sy;
                                                      pr_ysp1= pr_ysp1 * dx_s ; //dialog
koordinatenpunkt s'X'y+
                                                      pr_ysp1+=dx_sy;

                                             if(y_m<y_m_max)
                                             {
                                                    ooo.MoveTo (tmp_0*frmX+ posX,
y_m*frmY+y_m_1+ posY);
                                                    ooo.LineTo (pr_ysp0*frmX+ posX, y_m*frmY+
posY     );

                                                    ooo.MoveTo (tmp_1*frmX+ posX,
y_m*frmY+y_m_1+ posY);
                                                    ooo.LineTo (pr_ysp1*frmX+ posX, y_m*frmY +
posY     );
                                                    if(sRi_q)if(y_m<=y_m_min+y_m_1)
                                                    ooo.TextOut(pr_ysp1*frmX+3+ posX,y_m*frmY-
5+ posY,  "t·ôR(y)");// ôRy
                                             }
                                                           tmp_0=pr_ysp0;
                                                           tmp_1=pr_ysp1;
                                      }
                              }
                      }


       ////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////
                      // THETA Fp Achsen zeichnen
                      //

                      if(sw_mod_==3)if(am_) // F(p) am linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_am); //textfarbe
                                                          CString cr_="x";
                              ooo.SelectObject(&op1);          if(am_q==1)cr_+="=";
                                                          if(am_q==1)cr_+=ftoc(qY.am,3);

                              if(sw_inv==0) //F(p)
                              {
                                     float pr_yam  = (qY.am +sc0y)/(max_y+sc0y); //qam
proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qam y
                                              pr_yam-=dy_sy;

                                             ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                             ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(am_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  am wert y
                              }

                              if(sw_inv==1)//F-1(p)
                              {
                                     float pr_xam  = (qY.am +sc0x)/(max_x+sc0x); //qam
proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qam x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(am_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-10+
posY,  cr_);//  am wert x
                              }
                      }

                      if(sw_mod_==3)if(sd_) // F(p) sd linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_sd); //textfarbe
                                                                CString cr_="□·s";
                              ooo.SelectObject(&op2);          if(sd_q==1)cr_+="= ";
                                                          if(sd_q==1)cr_+=ftoc(qY.sd*ci_zp,3);

                              if(sw_inv==0) //F(p)
                              {
                                     float pr_yam  = (qY.am+(qY.sd)*ci_zp +sc0y)/(max_y+sc0y);
//qsd   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qsd+ y
                                              pr_yam-=dy_sy;

                                             ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                             ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(sd_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  sd wert y

                                              pr_yam  = (qY.am-(qY.sd)*ci_zp
+sc0y)/(max_y+sc0y); //qsd   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qsd- y
                                              pr_yam-=dy_sy;

                                             ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                             ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(sd_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  sd wert y
                              }

                              if(sw_inv==1)//F-1(p)
                              {
                                     float pr_xam  = (qY.am+(qY.sd)*ci_zp +sc0x)/(max_x+sc0x);
//qsd   proportion
```

```
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qsd+ x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(sd_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-10+
posY,  cr_);//  sd wert x

                                              pr_xam  = (qY.am-(qY.sd)*ci_zp
+sc0x)/(max_x+sc0x); //qsd   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qsd- x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(sd_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-10+
posY,  cr_);//  sd wert x
                              }
                      }

                      if(sw_mod_==3)if(sdg_) // F(p) sd' linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_sdg); //textfarbe
                                                                  CString cr_="□·ô";
                              ooo.SelectObject(&op4);          if(sdg_q==1)cr_+="= ";

if(sdg_q==1)cr_+=ftoc(qY.sdg*ci_zp,3);

                              if(sw_inv==0) //F(p)
                              {
                                     float pr_yam  = (qY.am+(qY.sdg)*ci_zp
+sc0y)/(max_y+sc0y); //qsd'   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qsd'+ y
                                              pr_yam-=dy_sy;

                                              ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                              ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(sdg_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  ô wert y

                                              pr_yam  = (qY.am-(qY.sdg)*ci_zp
+sc0y)/(max_y+sc0y); //qsd'   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qsd'- y
                                              pr_yam-=dy_sy;

                                              ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                              ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(sdg_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  ô wert y
                              }

                              if(sw_inv==1)//F-1(p)
                              {
                                     float pr_xam  = (qY.am+(qY.sdg)*ci_zp
+sc0x)/(max_x+sc0x); //qsd'   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qsd+ x
                                              pr_xam+=dx_sy;

                                              ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                              ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(sdg_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-10+
posY,  cr_);//  ô wert x
```

```
                                              pr_xam  = (qY.am-(qY.sdg)*ci_zp
+sc0x)/(max_x+sc0x); //qsd'   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qsd'- x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(sdg_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-10+
posY,  cr_);//  ô wert x
                              }
                      }

                      if(sw_mod_==3)if(sgam_) // F(p) s'am linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_sgam); //textfarbe
                                                                  CString cr_="t·ôx";
                              ooo.SelectObject(&op3);           if(sgam_q==1)cr_+="= ";

if(sgam_q==1)cr_+=ftoc(qY.sgam*ci_tp,3);
                              if(sw_inv==0) //F(p)
                              {
                                     float pr_yam  = (qY.am+(qY.sgam)*ci_tp
+sc0y)/(max_y+sc0y); //qs'am   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qs'am + y
                                              pr_yam-=dy_sy;

                                               ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                                      ooo.LineTo (x_m_max*frmX+ posX,
pr_yam*frmY+ posY);

                                              pr_yam  = (qY.am-(qY.sgam)*ci_tp
+sc0y)/(max_y+sc0y); //qs'am   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qs'am - y
                                              pr_yam-=dy_sy;

                                               ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                               ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(sgam_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  ôx wert y
                              }

                              if(sw_inv==1)//F-1(p)
                              {
                                     float pr_xam  = (qY.am+(qY.sgam)*ci_tp
+sc0x)/(max_x+sc0x); //qs'am   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qs'am+ x
                                              pr_xam+=dx_sy;

                                               ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                               ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);

                                              pr_xam  = (qY.am-(qY.sgam)*ci_tp
+sc0x)/(max_x+sc0x); //qs'am   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qs'am - x
                                              pr_xam+=dx_sy;

                                               ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                               ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(sgam_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-
10+ posY,  cr_);//  ôx wert x
                              }
                      }
                      if(sw_mod_==3)if(a3_) // F(p) a3 linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_a3); //textfarbe
                                                              CString cr_="a3";
                              ooo.SelectObject(&op5);       if( a3_q==1)cr_+="= ";
                                                       if( a3_q==1)cr_+=ftoc(qY.a3,3);
                                                       if( a3_q==1)cr_+="z";

                              if(sw_inv==0) //F(p)
                              {
                                     float pr_yam  = ((qY.am+qY.a3*qY.sd) +sc0y)/(max_y+sc0y);
//qa3   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qa3 y
                                              pr_yam-=dy_sy;

                                             ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                             ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(a3_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  a3 wert y
                              }

                              if(sw_inv==1)//F-1(p)
                              {
                                     float pr_xam  = ((qY.am+qY.a3*qY.sd) +sc0x)/(max_x+sc0x);
//qa3   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qa3 x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(a3_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-19+
posY,  cr_);//  a3 wert x
                              }
                      }

                      if(sw_mod_==3)if(a4_) // F(p) a4 linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_a4); //textfarbe
                                                                CString cr_="a4";
                              ooo.SelectObject(&op8);          if(a4_q==1)cr_+="= ";
                                                          if(a4_q==1)cr_+=ftoc(qY.a4,3);
                                                          if(a4_q==1)cr_+="z";

                              if(sw_inv==0) //F(p)
                              {
                                     float pr_yam  = ((qY.am+qY.a4*qY.sd) +sc0y)/(max_y+sc0y);
//qa4   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qa4+ y
                                              pr_yam-=dy_sy;

                                             ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                             ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);

                                              pr_yam  = ((qY.am-qY.a4*qY.sd)
+sc0y)/(max_y+sc0y); //qa4   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qa4- y
                                              pr_yam-=dy_sy;

                                             ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                             ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(a4_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  a4 wert y
                              }
                              if(sw_inv==1)//F-1(p)
                              {
                                     float pr_xam  = ((qY.am+qY.a4*qY.sd) +sc0x)/(max_x+sc0x);
//qa4   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qa4+ x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);

                                              pr_xam  = ((qY.am-qY.a4*qY.sd)
+sc0x)/(max_x+sc0x); //qa4   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qa4- x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(a4_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-19+
posY,  cr_);//  a4 wert x
                              }
                      }

                      if(sw_mod_==3)if(sga3_) // F(p) sga3 linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_sga3); //textfarbe
                                                                 CString cr_="t·ôa3";
                              ooo.SelectObject(&op7);           if(sa3g_q==1)cr_+="= ";

if(sa3g_q==1)cr_+=ftoc(qY.sga3*ci_tp,3);
                                                          if(sa3g_q==1)cr_+="z";

                              if(sw_inv==0) //F(p)
                              {
                                     //float pr_yam  = ((qY.am+(qY.sga3*ci_t)*qY.sd)
+sc0y)/(max_y+sc0y); //qs'a3   proportion um am
                                     float pr_yam  =
(((qY.am+qY.a3*qY.sd)+(qY.sga3*ci_tp)*qY.sd) +sc0y)/(max_y+sc0y); //qs'a3   proportion um a3
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qs'a3 + y
                                              pr_yam-=dy_sy;

                                               ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                               ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);

                                              //pr_yam  = ((qY.am-
(qY.sga3*ci_t)*qY.sd)+sc0y)/(max_y+sc0y); //qs'a3   proportion um am
                                              pr_yam  = (((qY.am+qY.a3*qY.sd)-
(qY.sga3*ci_tp)*qY.sd)+sc0y)/(max_y+sc0y); //qs'a3   proportion um a3

                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qs'a3 - y
                                              pr_yam-=dy_sy;

                                               ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                               ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(sa3g_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  ôa3 wert y
                              }

                              if(sw_inv==1)//F-1(p)
                              {
                                     //float pr_xam  = ((qY.am+(qY.sga3*ci_t)*qY.sd)
+sc0x)/(max_x+sc0x); //qs'a3   proportion um am
                                     float pr_xam  =
(((qY.am+qY.a3*qY.sd)+(qY.sga3*ci_tp)*qY.sd) +sc0x)/(max_x+sc0x); //qs'a3   proportion um a3
```

```
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qs'a3+ x
                                              pr_xam+=dx_sy;

                                               ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                               ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(sa3g_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-
19+ posY,  cr_);//  ôa3 wert x

                                              //pr_xam  = ((qY.am-(qY.sga3*ci_t)*qY.sd)
+sc0x)/(max_x+sc0x); //qs'a3   proportion um am
                                              pr_xam  = (((qY.am+qY.a3*qY.sd)-
(qY.sga3*ci_tp)*qY.sd) +sc0x)/(max_x+sc0x); //qs'a3   proportion um a3
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qs'a3 - x
                                              pr_xam+=dx_sy;

                                     ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+ posY);
                                     ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+ posY);
                              }
                      }

                      if(sw_mod_==3)if(sga4_) // F(p) sga4 linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_sga4); //textfarbe
                                                                      CString cr_="t·ôa4";
                              ooo.SelectObject(&op10);         if(sa4g_q==1)cr_+="= ";

if(sa4g_q==1)cr_+=ftoc(qY.sga4*ci_tp,3);
                                                              if(sa4g_q==1)cr_+="z";
                              if(sw_inv==0) //F(p)
                              {
                                     //float pr_yam  = ((qY.am+(qY.sga4*ci_t)*qY.sd)
+sc0y)/(max_y+sc0y); //qs'a4   proportion um am
                                     float pr_yam  =
(((qY.am+qY.a4*qY.sd)+(qY.sga4*ci_tp)*qY.sd) +sc0y)/(max_y+sc0y); //qs'a4   proportion um a4
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qs'a4 + y
                                              pr_yam-=dy_sy;

                                               ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                               ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);

                                              //pr_yam  = ((qY.am-(qY.sga4*ci_t)*qY.sd)
+sc0y)/(max_y+sc0y); //qs'a4   proportion um am
                                              pr_yam  = (((qY.am+qY.a4*qY.sd)-
(qY.sga4*ci_tp)*qY.sd) +sc0y)/(max_y+sc0y); //qs'a4   proportion um a4
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qs'a4 - y
                                              pr_yam-=dy_sy;

                                               ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                               ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(sa4g_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  ôa4 wert y
                              }

                              if(sw_inv==1)//F-1(p)
                              {
                                     //float pr_xam  = ((qY.am+(qY.sga4*ci_t)*qY.sd)
+sc0x)/(max_x+sc0x); //qs'a4   proportion um am
                                     float pr_xam  =
(((qY.am+qY.a4*qY.sd)+(qY.sga4*ci_tp)*qY.sd) +sc0x)/(max_x+sc0x); //qs'a4   proportion um a4
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qs'a4+ x
                                              pr_xam+=dx_sy;

                                               ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                               ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
```

```
                                     if(sa4g_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-
19+ posY,  cr_);//  ôa4 wert x

                                              //pr_xam  = ((qY.am-(qY.sga4*ci_t)*qY.sd)
+sc0x)/(max_x+sc0x); //qs'a4   proportion um am
                                              pr_xam  = (((qY.am+qY.a4*qY.sd)-
(qY.sga4*ci_tp)*qY.sd) +sc0x)/(max_x+sc0x); //qs'a4   proportion um a4
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qs'a4 - x
                                              pr_xam+=dx_sy;

                                     ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+ posY);
                                     ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+ posY);
                              }
                      }

                   if(sw_mod_==3)if(ag3_) // F(p) ag3 linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_ag3); //textfarbe
                                                                 CString cr_="â3";
                              ooo.SelectObject(&op6);          if(ag3_q==1)cr_+="= ";
                                                          if(ag3_q==1)cr_+=ftoc(qY.ag3,3);
                                                          if(ag3_q==1)cr_+="z";
                              if(sw_inv==0) //F(p)
                              {
                                     float pr_yam  = ((qY.am+qY.ag3*qY.sd)
+sc0y)/(max_y+sc0y); //qa3'   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qa3' y
                                              pr_yam-=dy_sy;

                                              ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                              ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(ag3_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  â3 wert y
                              }

                              if(sw_inv==1)//F-1(p)
                              {
                                     float pr_xam  = ((qY.am+qY.ag3*qY.sd)
+sc0x)/(max_x+sc0x); //qa3'   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qa3' x
                                              pr_xam+=dx_sy;

                                              ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                              ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(ag3_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-19+
posY,  cr_);//  â3 wert x
                              }
                      }

                      if(sw_mod_==3)if(ag4_) // F(p) ag4 linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_ag4); //textfarbe
                                                                 CString cr_="â4";
                              ooo.SelectObject(&op9);          if(ag4_q==1)cr_+="= ";
                                                          if(ag4_q==1)cr_+=ftoc(qY.ag4,3);
                                                          if(ag4_q==1)cr_+="z";
                              if(sw_inv==0) //F(p)
                              {
                                     float pr_yam  = ((qY.am+qY.ag4*qY.sd)
+sc0y)/(max_y+sc0y); //qa4'   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qa4'+ y
                                              pr_yam-=dy_sy;

                                              ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                              ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
```

```
                                               pr_yam  = ((qY.am-qY.ag4*qY.sd)
+sc0y)/(max_y+sc0y); //qa4'   proportion
                                               pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qa4'- y
                                               pr_yam-=dy_sy;

                                              ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                              ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(ag4_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  â4 wert y
                              }

                              if(sw_inv==1)//F-1(p)
                              {
                                     float pr_xam  = ((qY.am+qY.ag4*qY.sd)
+sc0x)/(max_x+sc0x); //qa4'   proportion
                                               pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qa4'+ x
                                               pr_xam+=dx_sy;

                                              ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                              ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);

                                               pr_xam  = ((qY.am-qY.ag4*qY.sd)
+sc0x)/(max_x+sc0x); //qa4'   proportion
                                               pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qa4'- x
                                               pr_xam+=dx_sy;

                                              ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                              ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(ag4_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-19+
posY,  cr_);//  â4 wert x
                              }
                      }


       ///////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////
                      // THETA Fe Achsen zeichnen
                      //

                      if(sw_mod_==4)if(x0_) // F(e) µ0 linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_am); //textfarbe
                                                           CString cr_="µ0";
                              ooo.SelectObject(&op13);       if(x0_q==1)cr_+="=";
                                                           if(x0_q==1)cr_+=ftoc(qX.am,3);

                              if(sw_inv==1) //F-1(e)
                              {
                                     float pr_yam  = (qX.am +sc0y)/(max_y+sc0y); //qam
proportion
                                               pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qam y
                                               pr_yam-=dy_sy;

                                             ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                             ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(x0_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  µ0 wert y
                              }

                              if(sw_inv==0)//F(e)
                              {
                                     float pr_xam  = (qX.am +sc0x)/(max_x+sc0x); //qam
proportion
                                               pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qam x
```

```
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(x0_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-10+
posY,  cr_);//  µ0 wert x
                              }
                      }

                      if(sw_mod_==4)if(x1_) // F(e) µ1 linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_am); //textfarbe
                                                             CString cr_="µ1";
                              ooo.SelectObject(&op14);        if(x1_q==1)cr_+="=";
                                                            if(x1_q==1)cr_+=ftoc(qY.am,3);

                              if(sw_inv==1) //F-1(e)
                              {
                                     float pr_yam  = (qY.am +sc0y)/(max_y+sc0y); //qam
proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qam y
                                              pr_yam-=dy_sy;

                                             ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                             ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(x1_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  µ1 wert y
                              }

                              if(sw_inv==0)//F(e)
                              {
                                     float pr_xam  = (qY.am +sc0x)/(max_x+sc0x); //qam
proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qam x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(x1_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-10+
posY,  cr_);//  µ1 wert x
                              }
                      }

                      if(sw_mod_==4)if(s0_) // F(e) s0 linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_s0); //textfarbe
                                                                  CString cr_="□·s0";
                              ooo.SelectObject(&op15);          if(s0_q==1)cr_+="= ";
                                                          if(s0_q==1)cr_+=ftoc(qX.sd*ci_ze,3);

                              if(sw_inv==1) //F-1(e)
                              {
                                     float pr_yam  = (qX.am+(qX.sd)*ci_ze +sc0y)/(max_y+sc0y);
//qs0   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qs0+ y
                                              pr_yam-=dy_sy;

                                              ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                              ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(s0_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  s0 wert y

                                              pr_yam  = (qX.am-(qX.sd)*ci_ze
+sc0y)/(max_y+sc0y); //qs0   proportion
```

```
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qs0- y
                                              pr_yam-=dy_sy;

                                              ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                              ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(s0_q) ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  s0 wert y
                              }

                              if(sw_inv==0)//F(e)
                              {
                                     float pr_xam  = (qX.am+(qX.sd)*ci_ze +sc0x)/(max_x+sc0x);
//qs0   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qs0+ x
                                              pr_xam+=dx_sy;

                                              ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                              ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(s0_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-10+
posY,  cr_);//  s0 wert x

                                              pr_xam  = (qX.am-(qX.sd)*ci_ze
+sc0x)/(max_x+sc0x); //qs0   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qs0 - x
                                              pr_xam+=dx_sy;

                                              ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                              ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(s0_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-10+
posY,  cr_);//  s0 wert x

                              }
                      }

                      if(sw_mod_==4)if(s1_) // F(e) s1 linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_s1); //textfarbe
                                                                CString cr_="□·s1";
                              ooo.SelectObject(&op16);        if(s1_q==1)cr_+="= ";
                                                         if(s1_q==1)cr_+=ftoc(qY.sd*ci_ze,3);

                              if(sw_inv==1) //F-1(e)
                              {
                                     float pr_yam  = (qY.am+(qY.sd)*ci_ze +sc0y)/(max_y+sc0y);
//qs1   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qs1+ y
                                              pr_yam-=dy_sy;

                                              ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                              ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(s0_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  s1 wert y

                                              pr_yam  = (qY.am-(qY.sd)*ci_ze
+sc0y)/(max_y+sc0y); //qs1   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qs1- y
                                              pr_yam-=dy_sy;

                                              ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                              ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(s0_q) ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  s0 wert y
```

```
                              }

                              if(sw_inv==0)//F(e)
                              {
                                     float pr_xam  = (qY.am+(qY.sd)*ci_ze +sc0x)/(max_x+sc0x);
//qs1   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qs1+ x
                                              pr_xam+=dx_sy;

                                              ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                              ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(s0_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-10+
posY,  cr_);//  s1 wert x

                                              pr_xam  = (qY.am-(qY.sd)*ci_ze
+sc0x)/(max_x+sc0x); //qs1   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qs1 - x
                                              pr_xam+=dx_sy;

                                              ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                              ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(s0_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-10+
posY,  cr_);//  s1 wert x
                              }
                       }


                       if(sw_mod_==4)if(xc_) // F(e) xcrit linie
                       {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_am); //textfarbe
                                                           CString cr_="xcrit";
                              ooo.SelectObject(&op12);      if(x_q==1)cr_+="=";
                                                           if(x_q==1)cr_+=ftoc(qE.sw,3);

                              if(sw_inv==1) //F-1(e)
                              {
                                     float pr_yam  = (qE.sw +sc0y)/(max_y+sc0y); //qsw
proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qam y
                                              pr_yam-=dy_sy;

                                             ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                             ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(x_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  xcrit wert y
                              }

                              if(sw_inv==0)//F(e)
                              {
                                     float pr_xam  = (qE.sw +sc0x)/(max_x+sc0x); //qsw
proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qam x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(x_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-20+
posY,  cr_);//  xcrit wert x
                              }
                       }

                       if(sw_mod_==4)if(e_) // F(e) e linie
                       {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_am); //textfarbe
```

```
                                                           CString cr_="e";
                              ooo.SelectObject(&op11);      if(e_q==1)cr_+="=";
                                                           if(e_q==1)cr_+=ftoc(qE.e,3);

                              if(sw_inv==1) //F-1(e)
                              {
                                     float pr_yam  = (qX.am +sc0y)/(max_y+sc0y); //qam
proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qam y
                                              pr_yam-=dy_sy;

                                     float pr_yam1  = (qY.am +sc0y)/(max_y+sc0y); //qam
proportion
                                              pr_yam1= dlg.y - ( pr_yam1 * dy_s ); //dialog
koordinatenpunkt qam y
                                              pr_yam1-=dy_sy;

                                     float pr_xam  = (0.5 +sc0x)/(max_x+sc0x); //0.5
proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qam x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX, pr_yam*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX, pr_yam1*frmY+
posY);
                                     if(e_q)ooo.TextOut(pr_xam*frmX+
posX,((pr_yam+pr_yam1)/2)*frmY-15+ posY,  cr_);//  e wert y
                              }

                              if(sw_inv==0)//F(e)
                              {
                                     float pr_xam  = (qX.am +sc0x)/(max_x+sc0x); //qam
proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qam x
                                              pr_xam+=dx_sy;

                                     float pr_xam1  = (qY.am +sc0x)/(max_x+sc0x); //qam
proportion
                                              pr_xam1= pr_xam1 * dx_s ; //dialog
koordinatenpunkt qam x
                                              pr_xam1+=dx_sy;

                                     float pr_yam  = (0.5 +sc0y)/(max_y+sc0y); //0.5
proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qam y
                                              pr_yam-=dy_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,pr_yam*frmY+ posY);
                                             ooo.LineTo (pr_xam1*frmX+ posX,pr_yam*frmY+
posY);
                                     if(e_q)ooo.TextOut(((pr_xam+pr_xam1)/2)*frmX+
posX,pr_yam*frmY-15+ posY,  cr_);//  e wert x
                              }
                      }



       ///////////////////////////////////////////////////////////////////////////////////
//////////////////////////
                      // ACHSEN Skalierung, Skalenwerte und Grid zeichnen
                      //

                      ooo.SelectObject(&o3s);

                      if(sw_xS==1)// x achsen skalen markierung
                      {

float dx= ((max_x-min_x)      /tlg_x);

float dx1=dx;
float x_m_1=((x_m_max-x_m_min)/tlg_x);
                         for(float x_m= x_m_min ;x_m<=x_m_max;x_m+=x_m_1 )
                         {
                                ooo.MoveTo (x_m*frmX+ posX,dapty*frmY-2+posSCx+
posY);//skalenmarkierung x
                                ooo.LineTo (x_m*frmX+ posX,dapty*frmY+2+posSCx+ posY);

                                if(sw_Grdx)//grid x
                                {    ooo.SelectObject(&ogx);
                                            ooo.LineTo (x_m*frmX+ posX,y_m_min*frmY+ posY);
                                }

                                if(sw_xSw==1)// skalenwerte x
                                {
                                       ooo.SelectObject(&of2);
                                       ooo.SetTextColor(Ax_fb);
                                       ooo.SetBkMode(TRANSPARENT);

                                       if(x_m== x_m_min) ooo.TextOut(x_m*frmX +posAXx+
posX,dapty*frmY +posAXy+ posY, ftoc(min_x,ds_xSw) );
                                       if(x_m!= x_m_min )
                                       {
                                              ooo.TextOut(x_m*frmX +posAXx+ posX,dapty*frmY
+posAXy+ posY,  ftoc((min_x)+dx1,ds_xSw));

                                            dx1+=dx;
                                       }
                                }
                         }
                 }

                 ooo.SelectObject(&o4s);

                 if(sw_yS==1)// y achsen skalen markierung
                 {

float dy= ((max_y-min_y)      /tlg_y);

float dy1=dy;



float y_m_1=((y_m_max-y_m_min)/tlg_y);
                         for(float y_m= y_m_min ;y_m<=y_m_max;y_m+=y_m_1 )
                         {
                                ooo.MoveTo (daptx*frmX-2+posSCy+ posX, y_m*frmY+
posY);//skalenmarkierung y
                                ooo.LineTo (daptx*frmX+2+posSCy+ posX, y_m*frmY+ posY);

                                if(sw_Grdy)//grid y
                                {
                                       ooo.SelectObject(&ogy);
                                       ooo.LineTo (x_m_max*frmX+ posX, y_m*frmY+ posY);
                                }

                                if(sw_ySw==1)// skalenwerte y
                                {
                                       ooo.SelectObject(&of1);
                                       ooo.SetTextColor(Ay_fb);
                                       ooo.SetBkMode(TRANSPARENT);

                                       if(y_m== y_m_min) ooo.TextOut(daptx*frmX +posAYx+
posX, y_m*frmY +posAYy+ posY,  ftoc((min_y+max_y)-min_y,ds_ySw));
                                       if(y_m!= y_m_min )
                                       {
                                              ooo.TextOut(daptx*frmX +posAYx+ posX, y_m*frmY
+posAYy+ posY,  ftoc((min_y+max_y)-((min_y)+dy1),ds_ySw));

                                                                   dy1+=dy;
                                       }
                                }
                         }
                 }

          }//sw_drw / rerendern //
```

```
///////////////////////////////////////////////////////////////////////////////////////////////
///////////////////
          // VEKTOREN und Vektor-Koordinaten zeichnen
          //

               //Vektorkoordinatenpunkte
                            float dvptx;
                            float dvpty;
               if(sw_v0==0)if(!sw_mkoord_V)//
               {
                                     dvptx = egx *(mv1x/(e_x/sc)); //dialog-vektor-
koordinatenpunkt x
                                  dvpty = egy *(mv1y/(e_y/sc)); //dialog-vektor-
koordinatenpunkt y
               }
               if(sw_v0==1)if(!sw_mkoord_V)//bei r(x,y) (0,0)
               {
                                     dvptx = (        ((((0)) +sc0x)/(max_x+sc0x))*
dx_s)+dx_sy; //dialog-vektor-koordinatenpunkt x0
                                     dvpty = ( dlg.y -((((0)) +sc0y)/(max_y+sc0y))* dy_s)-
dy_sy; //dialog-vektor-koordinatenpunkt y0
               }
               if(sw_v0==2)if(!sw_mkoord_V)//bei F(p) (0.5,y/2)
               {
                        if(sw_inv==0)dvptx = (                  ((((0.5))
+sc0x)/(max_x+sc0x))* dx_s)+dx_sy; //dialog-vektor-koordinatenpunkt x0.5
                        if(sw_inv==0)dvpty = ( dlg.y -(((((max_y+min_y)/2))
+sc0y)/(max_y+sc0y))* dy_s)-dy_sy; //dialog-vektor-koordinatenpunkt y/2
                        if(sw_inv==1)dvptx = (        (((((max_x+min_x)/2))
+sc0x)/(max_x+sc0x))* dx_s)+dx_sy; //dialog-vektor-koordinatenpunkt x/2
                        if(sw_inv==1)dvpty = (             dlg.y -((((0.5))
+sc0y)/(max_y+sc0y))* dy_s)-dy_sy; //dialog-vektor-koordinatenpunkt y0.5
               }
               if(sw_v0==3)if(!sw_mkoord_V)//bei f(x) (xmax,ymax)
               {
                                     dvptx = (         (((max_x))  +sc0x)/(max_x+sc0x))*
dx_s)+dx_sy; //dialog-vektor-koordinatenpunkt xmax
                                     dvpty = ( dlg.y -(((((max_y))) +sc0y)/(max_y+sc0y))*
dy_s)-dy_sy; //dialog-vektor-koordinatenpunkt ymax
               }
               if(sw_v0==4)if(!sw_mkoord_V)//bei F(e) (xcrit,pmax)
               {
                       if(sw_inv==0)dvptx = (         ((((qE.sw)) +sc0x)/(max_x+sc0x))*
dx_s)+dx_sy; //dialog-vektor-koordinatenpunkt xcrit
                       if(sw_inv==0)dvpty = ( dlg.y -(((((max_y)))+sc0y)/(max_y+sc0y))* dy_s)-
dy_sy; //dialog-vektor-koordinatenpunkt pmax
                       if(sw_inv==1)dvptx = (        (((((max_x))) +sc0x)/(max_x+sc0x))*
dx_s)+dx_sy; //dialog-vektor-koordinatenpunkt pmax
                       if(sw_inv==1)dvpty = (  dlg.y -((((qE.sw)) +sc0y)/(max_y+sc0y))* dy_s)-
dy_sy; //dialog-vektor-koordinatenpunkt xcrit
               }

               if(sw_mkoord_V)//bei manueller vektor koordinatenbestimmung
               {
                                     dvptx = (         (((mVx))  +sc0x)/(max_x+sc0x))*
dx_s)+dx_sy; //dialog-vektor-koordinatenpunkt x manuell
                                     dvpty = ( dlg.y -(((((mVy))) +sc0y)/(max_y+sc0y))* dy_s)-
dy_sy; //dialog-vektor-koordinatenpunkt     y manuell
               }

               ooo.SelectObject(&o1);
               if(sw_xV==1)
               {
                      ooo.MoveTo(        dvptx*frmX+ posX, 0);     // vektor x
                      ooo.LineTo(        dvptx*frmX+ posX, dlg.y);
               }

               ooo.SelectObject(&o2);
               if(sw_yV==1)
               {
                      ooo.MoveTo( 0,     dvpty*frmY+ posY);        // vektor y
                      ooo.LineTo( dlg.x, dvpty*frmY+ posY);
               }

               //Vektorkoordinatenwerte
               double  xy_x = (((( egx*(    mv1x/(e_x/sc)))-dx_sy)/dx_s)*(max_x+sc0x))-sc0x;
               //double  xy_y=  ((((dlg.y-(dlg.y/15.0))-
(egy*(mv1y/(e_y/sc)))+dy_sy)/dy_s)*(max_y+sc0y))-sc0y; //alternativ
               double  xy_y = (((( egy*((1-mv1y)/(e_y/sc)))-dy_sy)/dy_s)*(max_y+sc0y))-sc0y;

               if(sw_v0==1){             xy_x=0;                xy_y=0;}               //bei
r(x,y) (0,0)
               if(sw_v0==2)if(sw_inv==0){xy_x=0.5;              xy_y=(max_y+min_y)/2;} //bei
F(p) (0.5,y/2)
               if(sw_v0==2)if(sw_inv==1){xy_x=(max_x+min_x)/2;  xy_y=0.5;}             //bei
F(q) (x/2,0.5)
               if(sw_v0==3){             xy_x=max_x;            xy_y=max_y;}           //bei
f(x) (xmax,ymax)
               if(sw_v0==4)if(sw_inv==0){xy_x=qE.sw;            xy_y=max_y;}           //bei
F(e) (xcrit,pmax)
               if(sw_v0==4)if(sw_inv==1){xy_x=max_x;            xy_y=qE.sw;}           //bei
F(e) (pmax,xcrit)

                                         xy_x+=corx;            xy_y+=cory;
               //                       Vx_m=xy_x;            Vy_m=xy_y;
//global für koordinatenübergabe an koordinateneinstellungsdialog
               //if(!sw_mkoord_V){mVx=Vx_m;              mVy=Vy_m;     }                //nicht
bei manueller koordinateneinstellung
               if(!sw_mkoord_V){     mVx=xy_x;              mVy=xy_y;     } //Vx_m, Vy_m
überflüssig...

                                                                CString c="(";
               if(sw_inv==0)if(sw_xK==1)  { if(sw_mod_==1)c+="x:"; if(sw_mod_==2)c+="zx:";
if(sw_mod_==5)c+="zx:";if(sw_mod_==3)c+="p:";if(sw_mod_==4)c+="q:";}
               if(sw_inv==1)if(sw_xK==1)  { if(sw_mod_==1)c+="y:"; if(sw_mod_==2)c+="zx:";
if(sw_mod_==5)c+="zy:";if(sw_mod_==3)if(sw_pq==0)c+="x:";if(sw_mod_==3)if(sw_pq==1)c+="y:";if(
sw_mod_==4)c+="p:";}//f-1(x)
               if(sw_xK==1)if(!sw_mkoord_V)                c+=ftoc(xy_x,ds_xk);
               if(sw_xK==1)if( sw_mkoord_V)                c+=ftoc(mVx,ds_xk);
               if(sw_xK==1&&sw_yK==1)                      c+="; ";
               if(sw_inv==0)if(sw_yK==1)  { if(sw_mod_==1)c+="y:"; if(sw_mod_==2)c+="zy:";
if(sw_mod_==5)c+="zy:";if(sw_mod_==3)if(sw_pq==0)c+="x:";if(sw_mod_==3)if(sw_pq==1)c+="y:";if(
sw_mod_==4)c+="p:";}
               if(sw_inv==1)if(sw_yK==1)  { if(sw_mod_==1)c+="x:"; if(sw_mod_==2)c+="zy:";
if(sw_mod_==5)c+="zx:";if(sw_mod_==3)c+="p:";if(sw_mod_==4)c+="q:";} //f-1(x)
               if(sw_yK==1)if(!sw_mkoord_V)                c+=ftoc(xy_y,ds_yk);
               if(sw_yK==1)if( sw_mkoord_V)                c+=ftoc(mVy,ds_yk);
                                                                                c+=")";
               ooo.SelectObject(&ofv);
               ooo.SetTextColor(V_fb);
               ooo.SetBkMode(2);                                //OPAQUE koordinatenwerte vor
die funktion
               if(sw_xK||sw_yK)
               ooo.TextOut(dvptx*frmX+ posVx+ posX, dvpty*frmY +posVy+ posY,    c  );


////////////////////////////////////////////////////////////////////////////////////////////////
///////////////////

               if(sw_status)//statusleiste
               {
                      CFont ofs; ofs.CreateFont(14, 4,
0,0,400,0,0,0,0,OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,DEFAULT_PITCH,"Lucida
Sans Unicode"  );//Statustext Font
            CPen oln;  oln.CreatePen(PS_SOLID,     1,16777215); //linie weiss
                      CPen oln1;oln1.CreatePen(PS_SOLID,     1,8421504); //linie grau

                                 CRect rect(0, dlg.y-10,dlg.x, dlg.y); //fläche
                      ooo.FillSolidRect(rect,13357270);
                      ooo.SelectObject(&oln);

                      ooo.MoveTo( 0,     dlg.y-11) ;   //
                      ooo.LineTo( dlg.x, dlg.y-11);    //

                      ooo.MoveTo( dlg.x-10,     dlg.y-1) ;   //grip
                      ooo.LineTo( dlg.x-1,     dlg.y-10) ;
                      ooo.SelectObject(&oln1);
                      ooo.MoveTo( dlg.x-9,     dlg.y-1) ;   //
                      ooo.LineTo( dlg.x,     dlg.y-10) ;
                      ooo.MoveTo( dlg.x-8,     dlg.y-1) ;   //
                      ooo.LineTo( dlg.x,     dlg.y-9) ;
/*
                      ooo.SelectObject(&oln);
                      ooo.MoveTo( dlg.x-6,     dlg.y-1) ;   //
```

```
                       ooo.LineTo( dlg.x,     dlg.y-7) ;
                       ooo.SelectObject(&oln1);
                       ooo.MoveTo( dlg.x-5,     dlg.y-1) ;   //
                       ooo.LineTo( dlg.x,     dlg.y-6) ;
                       ooo.MoveTo( dlg.x-4,     dlg.y-1) ;   //
                       ooo.LineTo( dlg.x,     dlg.y-5) ;*/

                       ooo.SelectObject(&ofs); //
                       ooo.SetTextColor(1);
                       ooo.SetBkMode(0);
                                                CString stx;
                       if(!sw_inv)
                       {
                                                       if(sw_mod_==1) stx="f(x)";
                                                       if(sw_mod_==5) stx="f(zx)";
                                                       if(sw_mod_==2) stx="r(x,y)";
                                                       if(sw_mod_==3) stx="F(p)";
                                                       if(sw_mod_==4) stx="F(e)";
                       }

                       if(sw_inv)
                       {
                                                       if(sw_mod_==1) stx="f(y)";
                                                       if(sw_mod_==5) stx="f(zy)";
                                                       if(sw_mod_==2) stx="r(y,x)";
                                                       if(sw_mod_==3) stx="F(q)";
                                                       if(sw_mod_==4) stx="F(p)";
                       }

                       ooo.TextOut(2, dlg.y-12 ,        stx  ); //Statustext links

                                                        stx="n=";
                                                        stx+=itoc(n_);
                                                              if(sw_inv) stx+="  invers";
                       ooo.TextOut(dlg.x-100, dlg.y-12 ,stx   ); //Statustext rechts
               }
        }//else
}


HCURSOR CGRP2Dlg::OnQueryDragIcon(){ return (HCURSOR) m_hIcon; }

void CGRP2Dlg::OnSize(UINT nType, int cx, int cy)
{
        CDialog::OnSize(nType, cx, cy);

        if(cy>30) //bei mindestgrösse
        {
               if(cy>1)dlg.x=cx;
               if(cy>1)dlg.y=cy;

               if(!dynrnd)sw_drw=0;//dynamisches rendern und funktions-renderschalter

               sw_sz=1; //size marker

               posX=posY=0; //diagrammverschub nullsetzen
               frmX=frmY=1; //diagrammform ursprung

            RedrawWindow();
        }
}

void CGRP2Dlg::OnMouseMove(UINT nFlags, CPoint point) //
{
        if(csr_)//individuell-cursor schalter
        {
               if(sw_emf_in!=1)if(sw_csr==1)SetCursor(m_Csr1);
               if(sw_emf_in!=1)if(sw_csr==2)SetCursor(m_Csr2);
               if(sw_emf_in==1)             SetCursor(m_Csr3);
        }

        if(sw_sz){    sw_sz=0;sw_drw=1;       RedrawWindow();}

        if(sw_Vs)if(nFlags==MK_RBUTTON)  // vektoren verschieben
        {
               if(posX!=0||posY!=0||frmX!=1||frmY!=1)   SetCursor(0); //bei Diagramm shift

               sw_v0=0; if(!dynrnd)sw_drw=0;//dynamisches rendern und funktions-renderschalter
```

```
                sw_mkoord_V=0;//manueller Vektorenkoordinatenwert aus

                mv1x= 1.0*point.x/dlg.x;
                mv1y= 1.0*point.y/dlg.y;

                RedrawWindow();
        }

        if(sw_As)if(nFlags==MK_LBUTTON ) // achsen verschieben
        {
                if(posX!=0||posY!=0||frmX!=1||frmY!=1)   SetCursor(0); //bei Diagramm shift
                if(!dynrnd)sw_drw=0;//dynamisches rendern und funktions-renderschalter

                sw_mkoord_A=0;//manueller koordinatenwert aus

                mv2x=     1.0*point.x/dlg.x;
                mv2y=     1.0*point.y/dlg.y;

                RedrawWindow();
        }

        CDialog::OnMouseMove(nFlags, point);
}


BOOL CGRP2Dlg::OnMouseWheel(UINT nFlags, short zDelta, CPoint pt)
{
        if(nFlags==MK_RBUTTON )if(zDelta<0  ) cory+=0.01;
        if(nFlags==MK_RBUTTON )if(zDelta>=0 ) cory-=0.01;
        if(nFlags!=MK_RBUTTON )if(zDelta<0  ) corx+=0.01;
        if(nFlags!=MK_RBUTTON )if(zDelta>=0 ) corx-=0.01;

        RedrawWindow();

        return CDialog::OnMouseWheel(nFlags, zDelta, pt);
}

void CGRP2Dlg::OnLButtonDown(UINT nFlags, CPoint point)
{OnMouseMove(MK_LBUTTON,point);sw_bd=1;CDialog::OnLButtonDown(nFlags,
point);}//mouseschaltermarker
void CGRP2Dlg::OnRButtonDown(UINT nFlags, CPoint point)
{OnMouseMove(MK_RBUTTON,point);sw_bd=1;CDialog::OnRButtonDown(nFlags, point);}

void CGRP2Dlg::OnRButtonUp(UINT nFlags, CPoint point)
{
        sw_drw=1;//funktions-renderschalter

        sw_bd=0;//mouseschaltermarker

        RedrawWindow();

        CDialog::OnRButtonUp(nFlags, point);
}

void CGRP2Dlg::OnLButtonUp(UINT nFlags, CPoint point)
{
        sw_drw=1;//funktions-renderschalter

        sw_bd=0; //mouseschaltermarker
        RedrawWindow();

        CDialog::OnLButtonUp(nFlags, point);
}


void CGRP2Dlg::MINMAX() // THETA0 Funktionsmatrix einlesen min,max bestimmung//
{
        int dg=0, i_;
        char c1[40], c2[40];

        n_=0; i_=0;

             FILE *f_;
                          f_=fopen(filename,"r");
        do
        {
                fscanf(f_,"%s%s",      c1,          c2);
```

```
            if(sw_inv==0||(sw_inv==1&&sw_mod_==2)){ x_=atof(c1); y_=atof(c2);}
               if(sw_inv==1)          if(sw_mod_!=2) { x_=atof(c2); y_=atof(c1);} //f-1

                                             FVx_[i_]= x_              ;           //
Funktionsvektor (x)
                                                                     FVy_[i_]=               y_
;           // Funktionsvektor (y)
                                                                              i_++;

            if(dg==0)
               {
                      min_x = x_;  max_x = x_;
                   min_y = y_;  max_y = y_;
               }

               if(dg==1)
               {
                      if(x_ < min_x) min_x = x_;
                      if(x_ > max_x) max_x = x_;
                      if(y_ < min_y) min_y = y_;
                      if(y_ > max_y) max_y = y_;
               }

                 dg=1;

               n_++;

        }while (feof (f_) == 0);
                  fclose(f_);



        sc0x=0;
        sc0y=0;

        if(min_x<=0)sc0x=-min_x+0.3; //+1 randproportionen ...
        if(min_y<=0)sc0y=-min_y+0.3; //+1

                                                            // n_-=1; n korrektur
        CWinApp*  pApp = AfxGetApp();
                  pApp->WriteProfileInt("Funktion","N",ftoi(n_));
}

void CGRP2Dlg::fn_THETA_1() // deskriptivkennwerte //
{
        int i_=0;
        char c1[40], c2[40];

        typedef struct tagSUM{double x; double y; double x2; double y2; double xy;}SUM;
                                                                                    SUM q;

                                                 q.x =0;

                                                 q.y =0;

                                                 q.x2=0;

                                                 q.y2=0;

                                                        q.xy=0;
        FILE *f_;
                 f_=fopen(filename,"r");

        do
        {
               fscanf(f_,"%s%s",         c1,                   c2);
                            q.x+=atof(c1);        q.y+=atof(c2);
                                 q.x2+=pow(atof(c1),2);q.y2+=pow(atof(c2),2);
                                     q.xy+=atof(c1)*             atof(c2);

        }while (feof (f_) == 0);
                  fclose(f_);

        //THETA qX;
                   qX.sum=q.x;
                      qX.am= q.x/n_;
                      qX.sd= sqrt(q.x2/n_-pow(qX.am,2));
                      qX.sdg= sqrt(pow(qX.sd,2)*(n_/(n_-1)));
                      qX.sgam= sqrt((qX.sdg,2)/n_);

       //THETA qY;
                  qY.sum=q.y;
                      qY.am= q.y/n_;
                      qY.sd= sqrt(q.y2/n_-pow(qY.am,2));
                      qY.sdg= sqrt(pow(qY.sd,2)*(n_/(n_-1)));
                      qY.sgam= sqrt((qY.sdg,2)/n_);

                                                 qR=(q.xy/n_-qX.am*qY.am)/(qX.sd*qY.sd);
//Korrelationskoeffizient
                           qS = sqrt(1-pow(     qR,2));
//Standardvorhersagefehler
                           qsS= sqrt((n_-n_*pow(qR,2))/(n_-2));
//Geschätzter Standardvorhersagefehler
                      qsR= qsS /sqrt(n_);
//Geschätzter Korrelationsfehler


                      ci_zr=zp_funktion((100 -((100-CI_Pr)/2))/100);          //
Konfidenzintervall rxy
                      ci_tr=tp_funktion((100 -((100-CI_Pr)/2))/100,itof(n_-2)); //

                      ci_zp=zp_funktion((100 -((100-CI_Pp)/2))/100);          //
Konfidenzintervall Fp
                      ci_tp=tp_funktion((100 -((100-CI_Pp)/2))/100,itof(n_)); //

                      ci_ze=zp_funktion((100 -((100-CI_Pe)/2))/100);          //
Konfidenzintervall Fe
                      ci_te=tp_funktion((100 -((100-CI_Pe)/2))/100,itof(n_)); //
}

void CGRP2Dlg::fn_THETA_2() // z Transformation //
{
       int i_=0;
       char c1[40], c2[40];

       FILE *f_1;f_1=fopen(filename_z,"w");
       FILE *f_2;f_2=fopen(filename,"r");

       do
       {
               fscanf(f_2,"%s%s",           c1,                     c2);
               fprintf(f_1,"%f\t%f", (atof(c1)-qX.am)/qX.sd, (atof(c2)-qY.am)/qY.sd); //z-
Matrix
               if(i_<n_-1)fprintf(f_1,"\n"); //zeilenvorschub

                            FVx_[i_]= (atof(c1)-qX.am)/qX.sd                          ; //
Funktionsvektor (zx)
                                      FVy_[i_]=                         (atof(c2)-
qY.am)/qY.sd ; // Funktionsvektor (zy)
                                              i_++;

       }while (i_<n_);

       //fprintf(f_1,"%f\n",qR);

       fclose(f_1);
       fclose(f_2);
}

void CGRP2Dlg::fn_THETA_3(short sw) // sortierung, p(q) bestimmung //
{
       int i_=0;
       float i_f=0, pa1;
       char in_[33010][20];
       char c0[20];

       FILE *f_1;f_1=fopen(filename_p,"w");
       FILE *f_2;f_2=fopen(filename,"r");

       do
       {
               if(!sw)fscanf(f_2,"%s%s",   in_[i_],     c0);//x vektor sortieren
               if( sw)fscanf(f_2,"%s%s",   c0,     in_[i_]);//y vektor sortieren
```

```
                                                       strcpy(in_[i_],ftoc(atof(in_[i_]),9));
                                                                             i_++;
        }while (i_<n_);//feof (f_2) == 0
               fclose(f_2);

                                 fn_sort(in_,n_,0); //sortierungsfunktion

        f_2=fopen("~tmp_sort","r"); i_=0;

        do
        {
                                      fscanf(f_2,"%s", c0);
                                      pa1=(i_f+1)/n_;   //i_f+1 da n=1...M
                 if(p_p_sw==1||p_p_sw==2)if(pa1>=0.5)pa1=1-pa1;
                                          if(p_p_sw==2)pa1*=2;

               fprintf(f_1,"%s\t%s", ftoc(pa1,8),c0); //p-Matrix
               if(i_<n_-1)fprintf(f_1,"\n"); //zeilenvorschub

                                                 FVx_[i_]= pa1        ;          //
Funktionsvektor (p)
                                                 FVy_[i_]=          atof(c0) ;   //
Funktionsvektor (q)
                                                          i_++;
                                          i_f+=1;
        }while (i_f<n_);//feof (f_2) == 0

        fclose(f_2);

        fclose(f_1);
}

void CGRP2Dlg::fn_THETA_4() // verteilungskennwerte //
{
        char c1[40], c2[40];
        int i_=0;

        typedef struct tagSUM{double zx; double zy; double s3x; double s3y;
                                                     double s4x; double s4y;
                                                     double sum2x; double sum2y;
                                                     double sum3x; double sum3y;
                                                     double sum4x; double sum4y;}SUM;
                                                                                  SUM q;

                                               q.zx =0;

                                               q.zy =0;

                                               q.s3x=0;

                                               q.s3y=0;

                                                             q.s4x=0;

                                               q.s4y=0;

                                                             q.sum2x=0;

                                                             q.sum2y=0;

                                                             q.sum3x=0;

                                                             q.sum3y=0;

                                                             q.sum4x=0;

                                                             q.sum4y=0;
        FILE *f_;
                 f_=fopen(filename,"r");

        do
        {
               fscanf(f_,"%s%s",             c1,                           c2);
                                       q.zx=(atof(c1)-qX.am)/qX.sd;  q.zy=(atof(c2)-
qY.am)/qY.sd; //z-Werte
                               q.s3x+=pow(q.zx,3);           q.s3y+=pow(q.zy,3);
                               q.s4x+=pow(q.zx,4);           q.s4y+=pow(q.zy,4);
```

```
                              q.sum2x+=pow((atof(c1)-qX.am),2);q.sum2y+=pow((atof(c2)-
qY.am),2);
                              q.sum3x+=pow((atof(c1)-qX.am),3);q.sum3y+=pow((atof(c2)-
qY.am),3);
                              q.sum3x+=pow((atof(c1)-qX.am),4);q.sum4y+=pow((atof(c2)-
qY.am),4);

            i_++;
        }while (i_<=n_);
                  fclose(f_);


        //THETA qX;
                   qX.a3=q.s3x/n_;
                      qX.a4=q.s4x/n_-3;
                   qX.sga3=sqrt(6/n_);
                      qX.sga4=2*sqrt(6/n_);
                      qX.ag3=n_*q.sum3x/((n_-1)*(n_-2)*pow(qX.sdg,3));
                      qX.ag4=(n_*(n_+1)*q.sum4x-(3*q.sum2x)*q.sum2x*(n_-1))/((n_-1)*(n_-
2)*(n_-3)*pow(qX.sdg,4));

        //THETA qY;
                      qY.a3=q.s3y/n_;
                      qY.a4=q.s4y/n_-3;
                      qY.sga3=sqrt(6/n_);
                      qY.sga4=2*sqrt(6/n_);
                      qY.ag3=n_*q.sum3y/((n_-1)*(n_-2)*pow(qY.sdg,3));
                      qY.ag4=(n_*(n_+1)*q.sum4y-(3*q.sum2y)*q.sum2y*(n_-1))/((n_-1)*(n_-
2)*(n_-3)*pow(qY.sdg,4));
}

void CGRP2Dlg::fn_THETA_5() // verteilungskennwerte effektgrösse//
{
        int i_=0;
        float i_f=0, pa1;
        char in_[33010][20];
        char c0[20];

        FILE *f_1;f_1=fopen(filename_p,"w");
        FILE *f_2;f_2=fopen(filename,"r");

        do
        {
               fscanf(f_2,"%s%s",   in_[i_],     c0);//x0 vektor sortieren


                                       strcpy(in_[i_],ftoc(atof(in_[i_]),9));
                                                             i_++;
        }while (i_<n_);
            fclose(f_2);

                         fn_sort(in_,n_,0); //sortierungsfunktion


                   f_2=fopen("~tmp_sort","r"); i_=0;
        do
        {
                                 fscanf(f_2,"%s", c0);
                                              pa1=(i_f+1)/n_;
                      if(p_e_sw==1||p_e_sw==2)if(pa1>=0.5)pa1=1-pa1;
                                           if(p_e_sw==2)pa1*=2;
       fprintf(f_1,"%s\t%s\n", c0,ftoc(pa1,8)); //pa1-Matrix


                                    FVy_[i_]= pa1        ;       // Funktionsvektor (p)
                                       FVx_[i_]=       atof(c0) ;   // Funktionsvektor (x0)
                                                i_++;
                                                i_f+=1;

        }while (i_f<n_);//feof (f_2) == 0

        fclose(f_2);
               f_2=fopen(filename,"r");

                                                i_=0;


        do
        {
                fscanf(f_2,"%s%s",   c0,     in_[i_]);//x1 vektor sortieren


                                        strcpy(in_[i_],ftoc(atof(in_[i_]),9));
                                                                                 i_++;
        }while (i_<n_);//feof (f_2) == 0
            fclose(f_2);

                                   fn_sort(in_,n_,0); //sortierungsfunktion

                        f_2=fopen("~tmp_sort","r"); i_f=0;
        do
        {
                                    fscanf(f_2,"%s", c0);
                                                            pa1= pa1=(i_f+1)/n_;
                         if(p_e_sw==1||p_e_sw==2)if(pa1>=0.5)pa1=1-pa1;
                                                if(p_e_sw==2)pa1*=2;
                fprintf(f_1,"%s\t%s", c0,ftoc(pa1,8) ); //pa1-Matrix
                if(i_f<n_-1)fprintf(f_1,"\n"); //zeilenvorschub


                                      FVy_[i_]= pa1        ;        // Funktionsvektor (p)
                                      FVx_[i_]=        atof(c0) ;   // Funktionsvektor (x1)
                                                i_++;
                                        i_f+=1;
        }while (i_f<n_);//feof (f_2) == 0

        fclose(f_2);
        fclose(f_1);

        qE.e=  (qY.am-qX.am)/qX.sdg; //effektgrösse epsilon
        qE.sw=((qX.am+qY.am)/2)*((qY.am-qX.am)/sqrt(pow((qY.am-qX.am),2))); //schwellenwert
xcrit
}

void CGRP2Dlg::integral() //integral
{
        int i_=0, sw=1;
        char c1[40], c2[40];

        CString file_in,            file_out;
                    file_in="~~tmp_in"; file_out="~~tmp_out";

        float ix=0, x, d_;
                        d_=(max_x-min_x)/(n_-1);

        if(filename!=filename_tmp)  filename=filename_tmp;

        FILE *f_2;f_2=fopen(filename,"r");
        FILE *f_1;f_1=fopen(file_in,"w");

        for(int in=1;in<=int_n;in++)//über ordnung n
        {
                do
                {
                        fscanf(f_2,"%s%s",                c1,                c2);
                                                                           x=atof(c1);
ix+=atof(c2);
                        fprintf(f_1,"%s\t%s", ftoc(x+d_/2,8),  ftoc(ix*d_,8)   );   //Integral-
Matrix
                        if(i_<n_-1)fprintf(f_1,"\n" );  //zeilenvorschub

                                i_++;
                }while (i_<n_);
                 i_=0;

                fclose(f_1);
                fclose(f_2);

                if( sw){FILE *f_2;f_2=fopen(file_in,  "r");FILE *f_1;f_1=fopen(file_out,"w");}
                if(!sw){FILE *f_2;f_2=fopen(file_out, "r");FILE *f_1;f_1=fopen(file_in, "w");}
                    sw=!sw;
        }
        fclose(f_1);
        fclose(f_2);

        filename_tmp=filename;
              if(!sw)filename=file_in;
```

```
                 if( sw)filename=file_out;

        MINMAX();
        //fn_THETA_1();

        RedrawWindow();
};

void CGRP2Dlg::differential() //differential
{
        int i_=0, sw=1;
        char c1[40], c2[40];

        CString file_in,            file_out;
                   file_in="~~tmp_in"; file_out="~~tmp_out";

        float dx=0, x, d_;
                      d_=(max_x-min_x)/(n_-1);

        if(filename!=filename_tmp)  filename=filename_tmp;

        FILE *f_2;f_2=fopen(filename,"r");
        FILE *f_1;f_1=fopen(file_in,"w");

        for(int in=1;in<=diff_n;in++)//über ordnung n
        {
            fscanf(f_2,"%s%s",                      c1,           c2);
                                                             dx=atof(c2);
               do
               {
                      fscanf(f_2,"%s%s",                   c1,           c2);
                                                                           x=atof(c1);
                      fprintf(f_1,"%s\t%s", ftoc(x-d_/2,8),  ftoc((atof(c2)-dx)/d_,8));
//Differential-Matrix

                         dx=atof(c2);
                      if(i_<n_-1)fprintf(f_1,"\n" ); //zeilenvorschub

                              i_++;
               }while (i_<n_);
                       i_=0;

               fclose(f_1);
               fclose(f_2);

               if( sw){FILE *f_2;f_2=fopen(file_in,  "r");FILE *f_1;f_1=fopen(file_out,"w");}
               if(!sw){FILE *f_2;f_2=fopen(file_out, "r");FILE *f_1;f_1=fopen(file_in, "w");}
                   sw=!sw;
        }
        fclose(f_1);
        fclose(f_2);

        filename_tmp=filename;
              if(!sw)filename=file_in;
                 if( sw)filename=file_out;

        MINMAX();
        //fn_THETA_1();

        RedrawWindow();
};

void CGRP2Dlg::kgl() //kurvenglättungs mittelung
{
        int i_=0;
        char c1[40], c2[40];

        CString file_in,            file_out;
                   file_in="~~tmp_in"; file_out="~~tmp_out";

        CString cx;
        CString stxt;

        float dx=0, dy=0,x, x1,x2;

        m_StatusDlg.Create(IDD_GRP2_STATUS);//statusfenster
        m_StatusDlg.ShowWindow(SW_SHOW);
        if(filename!=filename_tmp)  filename=filename_tmp;

        FILE *f_2;f_2=fopen(filename,"r");fscanf(f_2,"%s%s",         c1,             c2);
           fclose(f_2);                                                        x1=atof(c2);

                  f_2=fopen(filename,"r");

        FILE *f_1;f_1=fopen(file_in, "w");


        for(int in=1;in<=kgl_i;in++)//über iterationen i
        {
                                         stxt="Iteration i=";//statusfenster
                                         stxt+=itoc(in);
               m_StatusDlg.SetWindowText(stxt);

                                                         dx=x1;      //erster wert
               fscanf(f_2,"%s%s",         c1,             c2);
                                   cx=ctoc(c1);
               do
               {
                       fscanf(f_2,"%s%s",      c1,             c2);

                       fprintf(f_1,"%s\t%s\n", cx,  ftoc((atof(c2)+dx)/2,8));  //gemittelte
funktions-Matrix

               dx=atof(c2);
                                     cx=ctoc(c1);
                              i_++;
               }while (i_<n_);
                       i_=0;

               fprintf(f_1,"%s\t%s",       cx,            ftoc(dx,8));     //letzter wert

               fclose(f_1);
               fclose(f_2);

               f_2=fopen(file_in, "r"); f_1=fopen(file_out,"w");


               fscanf(f_2,"%s%s",         c1,             c2);
               fprintf(f_1,"%s\t%s\n",     c1,     ftoc(x1,8));

                                                   dx=atof(c2);
               do
               {
                       fscanf(f_2,"%s%s",      c1,             c2);

                       fprintf(f_1,"%s\t%s\n", c1,  ftoc((atof(c2)+dx)/2,8));  //gemittelte
funktions-Matrix

               dx=atof(c2);
                              i_++;
               }while (i_<n_-1);
                       i_=0;

               fscanf(f_2,"%s%s",         c1,             c2);
               fprintf(f_1,"%s\t%s",       c1,             c2);           //letzter wert

               fclose(f_1);
               fclose(f_2);

            f_2=fopen(file_out, "r"); f_1=fopen(file_in,"w");
        }

        do
        {
               fscanf(f_2,"%s%s",         c1,             c2);
               fprintf(f_1,"%s\t%s",       c1,             c2);
               if(i_<n_-1)fprintf(f_1,"\n" ); //zeilenvorschub

                       i_++;
        }while (i_<n_);

        fclose(f_1);
        fclose(f_2);

        filename_tmp=filename;
```

```
                    filename=file_in;

        MINMAX();
        //fn_THETA_1();

        m_StatusDlg.DestroyWindow();//statusfenster schliessen

        RedrawWindow();
};

void CGRP2Dlg::fx_0()//funktionsmatrix reset
{
        filename=filename_tmp;
        MINMAX();
        fn_THETA_1();

        RedrawWindow();

        if(sw_Theta){sw_Theta=0;OnAnsichtThetafensterq(1);} //Theta Ansicht aktualisieren
        if(sw_Fxy){sw_Fxy=0;OnAnsichtFunktionsmatrixfxy(1);} //Fxy Ansicht aktualisieren
};

void CGRP2Dlg::fx_1()//funktionsmatrix neu setzen
{
        if(filename!=filename_tmp)
        {
                FILE *inStream, *outStream;

                char c_1, c_2 ;

                int index = 1;

                inStream = fopen( filename, "rb" );

                                        if(file_ind==100)file_ind=1;//bei index==100
                         CString file;
                                 file=itoc(file_ind);               //funktionsmatrix
dateiindex setzen
                                                            file_ind++;
//dateiindex

                                           file+="_";

        if(file_ind>2&&file_ind<=10)filename_tmp=filename_tmp.Mid(2); //löschung von altem
dateiindex
                              if(file_ind> 10)filename_tmp=filename_tmp.Mid(3);
                                           file+=filename_tmp;
                outStream = fopen(file, "wb" );

                do //funktionsmatrix datei kopieren
                {
                                                                          c_1 =
fgetc(inStream);
                      if (index > 1) fputc (c_2,             outStream);
                                                             c_2 = c_1;
                              index++;

                }while (feof (inStream) == 0);

                fclose( inStream  );
                fclose( outStream );

                filename_tmp=file; //funktionsmatrix dateinamen neu setzen

                if(sw_Theta){sw_Theta=0;OnAnsichtThetafensterq(1);} //Theta Ansicht
aktualisieren
                if(sw_Fxy){sw_Fxy=0;OnAnsichtFunktionsmatrixfxy(1);} //Fxy Ansicht
aktualisieren

                Fenstertext(file); //
        }
};


void CGRP2Dlg::OnTimer(UINT nIDEvent)
{
        if(nIDEvent==2){sw_sz=0;}//rerender sizemarker switch ereignis
        if(nIDEvent==1&&sw_sz==0&&sw_drw==0&&sw_bd==0) {sw_drw=1;
if(dlg.y>30)RedrawWindow();}//rerender ereignis

        if(!sw_emf)if(nIDEvent==0)//haupt ereignisse
        {

           short rdw=0;
                if(sw_x!=sw_x_0){         sw_x_0=sw_x;                rdw=1;}
                if(sw_y!=sw_y_0){         sw_y_0=sw_y;                rdw=1;}
                if(sw_xm!=sw_xm_0){       sw_xm_0=sw_xm;              rdw=1;}
                if(sw_ym!=sw_ym_0){       sw_ym_0=sw_ym;              rdw=1;}
                if(sw_xA!=sw_xA_0){       sw_xA_0=sw_xA;              rdw=1;}
                if(sw_yA!=sw_yA_0){       sw_yA_0=sw_yA;              rdw=1;}
                if(sw_xV!=sw_xV_0){       sw_xV_0=sw_xV;              rdw=1;}
                if(sw_yV!=sw_yV_0){       sw_yV_0=sw_yV;              rdw=1;}
                if(sw_xS!=sw_xS_0){       sw_xS_0=sw_xS;              rdw=1;}
                if(sw_yS!=sw_yS_0){       sw_yS_0=sw_yS;              rdw=1;}
                if(sw_xSw!=sw_xSw_0){     sw_xSw_0=sw_xSw;            rdw=1;}
                if(sw_ySw!=sw_ySw_0){     sw_ySw_0=sw_ySw;            rdw=1;}
                if(sw_xK!=sw_xK_0){       sw_xK_0=sw_xK;              rdw=1;}
                if(sw_yK!=sw_yK_0){       sw_yK_0=sw_yK;              rdw=1;}
                if(sw_FK!=sw_FK_0){       sw_FK_0=sw_FK;              rdw=1;}
                if(sw_xy!=sw_xy_0){       sw_xy_0=sw_xy;              rdw=1;}
                if(corx!=corx0){          corx0=corx;                 rdw=1;}
                if(cory!=cory0){          cory0=cory;                 rdw=1;}
                if(tlg_x!=tlg_x_0){       tlg_x_0=tlg_x;              rdw=1;}
                if(tlg_y!=tlg_y_0){       tlg_y_0=tlg_y;              rdw=1;}
                if(ds_x!=ds_x_0){         ds_x_0=ds_x;                rdw=1;}
                if(ds_y!=ds_y_0){         ds_y_0=ds_y;                rdw=1;}
                if(ds_xk!=ds_xk_0){       ds_xk_0=ds_xk;              rdw=1;}
                if(ds_yk!=ds_yk_0){       ds_yk_0=ds_yk;              rdw=1;}
                if(ds_xSw!=ds_xSw_0){     ds_xSw_0=ds_xSw;            rdw=1;}
                if(ds_ySw!=ds_ySw_0){     ds_ySw_0=ds_ySw;            rdw=1;}
                if(sc!=sc_0){             sc_0=sc;                    rdw=1;}
                if(scy!=scy_0){           scy_0=scy;                  rdw=1;}
                if(fb_hg!=fb_hg_0){       fb_hg_0=fb_hg;              rdw=1;}
                if(fb_K!=fb_K_0){         fb_K_0=fb_K;                rdw=1;}
                if(fb_P!=fb_P_0){         fb_P_0=fb_P;                rdw=1;}
                if(linB_Fn!=linB_Fn_0){   linB_Fn_0=linB_Fn;          rdw=1;}
                if(linB_FnP!=linB_FnP_0){ linB_FnP_0=linB_FnP;        rdw=1;}
                if(fn_x_fon!=fn_x_fon_0){ fn_x_fon_0=fn_x_fon;        rdw=1;}
                if(fn_x_fb!=fn_x_fb_0){   fn_x_fb_0=fn_x_fb;          rdw=1;}
                if(fn_x_H!=fn_x_H_0){     fn_x_H_0=fn_x_H;            rdw=1;}
                if(fn_x_W!=fn_x_W_0){     fn_x_W_0=fn_x_W;            rdw=1;}
                if(fn_y_fon!=fn_y_fon_0){ fn_y_fon_0=fn_y_fon;        rdw=1;}
                if(fn_y_fb!=fn_y_fb_0){   fn_y_fb_0=fn_y_fb;          rdw=1;}
                if(fn_y_H!=fn_y_H_0){     fn_y_H_0=fn_y_H;            rdw=1;}
                if(fn_y_W!=fn_y_W_0){     fn_y_W_0=fn_y_W;            rdw=1;}
                if(Ax_fon!=Ax_fon_0){     Ax_fon_0=Ax_fon;            rdw=1;}
                if(Ax_fb!=Ax_fb_0){       Ax_fb_0=Ax_fb;              rdw=1;}
                if(Ax_H!=Ax_H_0){         Ax_H_0=Ax_H;                rdw=1;}
                if(Ax_W!=Ax_W_0){         Ax_W_0=Ax_W;                rdw=1;}
                if(Ay_fon!=Ay_fon_0){     Ay_fon_0=Ay_fon;            rdw=1;}
                if(Ay_fb!=Ay_fb_0){       Ay_fb_0=Ay_fb;              rdw=1;}
                if(Ay_H!=Ay_H_0){         Ay_H_0=Ay_H;                rdw=1;}
                if(Ay_W!=Ay_W_0){         Ay_W_0=Ay_W;                rdw=1;}
                if(V_fon!=V_fon_0){       V_fon_0=V_fon;              rdw=1;}
                if(V_fb!=V_fb_0){         V_fb_0=V_fb;                rdw=1;}
                if(V_H!=V_H_0){           V_H_0=V_H;                  rdw=1;}
                if(V_W!=V_W_0){           V_W_0=V_W;                  rdw=1;}
                if(mod_Fn!=mod_Fn_0){     mod_Fn_0=mod_Fn;            rdw=1;}
                if(mod_FnP!=mod_FnP_0){   mod_FnP_0=mod_FnP;          rdw=1;}
                if(linB_Ax!=linB_Ax_0){   linB_Ax_0=linB_Ax;          rdw=1;}
                if(linB_Ay!=linB_Ay_0){   linB_Ay_0=linB_Ay;          rdw=1;}
                if(linB_Vx!=linB_Vx_0){   linB_Vx_0=linB_Vx;          rdw=1;}
                if(linB_Vy!=linB_Vy_0){   linB_Vy_0=linB_Vy;          rdw=1;}
                if(mod_Ax!=mod_Ax_0){     mod_Ax_0=mod_Ax;            rdw=1;}
                if(mod_Ay!=mod_Ay_0){     mod_Ay_0=mod_Ay;            rdw=1;}
                if(mod_Vx!=mod_Vx_0){     mod_Vx_0=mod_Vx;            rdw=1;}
                if(mod_Vy!=mod_Vy_0){     mod_Vy_0=mod_Vy;            rdw=1;}
                if(fb_Ax!=fb_Ax_0){       fb_Ax_0=fb_Ax;              rdw=1;}
                if(fb_Ay!=fb_Ay_0){       fb_Ay_0=fb_Ay;              rdw=1;}
                if(fb_Vx!=fb_Vx_0){       fb_Vx_0=fb_Vx;              rdw=1;}
                if(fb_Vy!=fb_Vy_0){       fb_Vy_0=fb_Vy;              rdw=1;}
                if(CI_Pp!=CI_Pp_0){       CI_Pp_0=CI_Pp;              rdw=1;}
                if(CI_Pr!=CI_Pr_0){       CI_Pr_0=CI_Pr;              rdw=1;}
                if(CI_Pe!=CI_Pe_0){       CI_Pe_0=CI_Pe;              rdw=1;}
                if(ci_ze!=ci_ze_0){       ci_ze_0=ci_ze;              rdw=1;}
                if(rxy_D!=rxy_D_0){       rxy_D_0=rxy_D;             rdw=1;}
                if(rxy_!=rxy_0){          rxy_0=rxy_;                rdw=1;}
                if(ryx_!=ryx_0){          ryx_0=ryx_;                rdw=1;}
                if(sxy_!=sxy_0){          sxy_0=sxy_;                rdw=1;}
                if(syx_!=syx_0){          syx_0=syx_;                rdw=1;}
                if(s1xy_!=s1xy_0){        s1xy_0=s1xy_;              rdw=1;}
                if(s1yx_!=s1yx_0){        s1yx_0=s1yx_;              rdw=1;}
                if(srxy_!=srxy_0){        srxy_0=srxy_;              rdw=1;}
                if(sryx_!=sryx_0){        sryx_0=sryx_;              rdw=1;}
                if(srx_!=srx_0){          srx_0=srx_;                rdw=1;}
                if(sry_!=sry_0){          sry_0=sry_;                rdw=1;}
                if(a3_!=a3_0){            a3_0=a3_;                  rdw=1;}
                if(a4_!=a4_0){            a4_0=a4_;                  rdw=1;}
                if(ag3_!=ag3_0){          ag3_0=ag3_;                rdw=1;}
                if(ag4_!=ag4_0){          ag4_0=ag4_;                rdw=1;}
                if(sga3_!=sga3_0){        sga3_0=sga3_;              rdw=1;}
                if(sga4_!=sga4_0){        sga4_0=sga4_;              rdw=1;}
                if(am_!=am_0){            am_0=am_;                  rdw=1;}
                if(sd_!=sd_0){            sd_0=sd_;                  rdw=1;}
                if(sdg_!=sdg_0){          sdg_0=sdg_;                rdw=1;}
                if(sgam_!=sgam_0){        sgam_0=sgam_;              rdw=1;}
                if(e_!=e_0){              e_0=e_;                    rdw=1;}
                if(xc_!=xc_0){            xc_0=xc_;                  rdw=1;}
                if(x0_!=x0_0){            x0_0=x0_;                  rdw=1;}
                if(x1_!=x1_0){            x1_0=x1_;                  rdw=1;}
                if(s0_!=s0_0){            s0_0=s0_;                  rdw=1;}
                if(s1_!=s1_0){            s1_0=s1_;                  rdw=1;}
                if(sw_Grdx!=sw_Grdx_0){   sw_Grdx_0=sw_Grdx;         rdw=1;}
                if(sw_Grdy!=sw_Grdy_0){   sw_Grdy_0=sw_Grdy;         rdw=1;}
                if(mAx!=mAx_0){           mAx_0=mAx;                 rdw=1;}
                if(mAy!=mAy_0){           mAy_0=mAy;                 rdw=1;}
                if(mVx!=mVx_0){           mVx_0=mVx;                 rdw=1;}
                if(mVy!=mVy_0){           mVy_0=mVy;                 rdw=1;}
                if(posVx!=posVx_0){       posVx_0=posVx;             rdw=1;}
                if(posVy!=posVy_0){       posVy_0=posVy;             rdw=1;}
                if(posAXx!=posAXx_0){     posAXx_0=posAXx;           rdw=1;}
                if(posAXy!=posAXy_0){     posAXy_0=posAXy;           rdw=1;}
                if(posAYx!=posAYx_0){     posAYx_0=posAYx;           rdw=1;}
                if(posAYy!=posAYy_0){     posAYy_0=posAYy;           rdw=1;}
                if(posBXx!=posBXx_0){     posBXx_0=posBXx;           rdw=1;}
                if(posBXy!=posBXy_0){     posBXy_0=posBXy;           rdw=1;}
                if(posBYx!=posBYx_0){     posBYx_0=posBYx;           rdw=1;}
                if(posBYy!=posBYy_0){     posBYy_0=posBYy;           rdw=1;}
                if(posFXx!=posFXx_0){     posFXx_0=posFXx;           rdw=1;}
                if(posFXy!=posFXy_0){     posFXy_0=posFXy;           rdw=1;}
                if(posFYx!=posFYx_0){     posFYx_0=posFYx;           rdw=1;}
                if(posFYy!=posFYy_0){     posFYy_0=posFYy;           rdw=1;}
                if(posSCx!=posSCx_0){     posSCx_0=posSCx;           rdw=1;}
                if(posSCy!=posSCy_0){     posSCy_0=posSCy;           rdw=1;}

                if(posX!=posX_0){         posX_0=posX;               rdw=1;}
                if(posY!=posY_0){         posY_0=posY;               rdw=1;}

                if(frmX!=frmX_0){         frmX_0=frmX;               rdw=1;}
                if(frmY!=frmY_0){         frmY_0=frmY;               rdw=1;}

                if(mod_Grdx!=mod_Grdx_0){ mod_Grdx_0=mod_Grdx;       rdw=1;}
                if(mod_Grdy!=mod_Grdy_0){ mod_Grdy_0=mod_Grdy;       rdw=1;}
                if(fb_Grdx!=fb_Grdx_0){   fb_Grdx_0=fb_Grdx;         rdw=1;}
                if(fb_Grdy!=fb_Grdy_0){   fb_Grdy_0=fb_Grdy;         rdw=1;}
                if(linB_Grdx!=linB_Grdx_0){linB_Grdx_0=linB_Grdx;    rdw=1;}
                if(linB_Grdy!=linB_Grdy_0){linB_Grdy_0=linB_Grdy;    rdw=1;}
                if(linB_r!=linB_r_0){     linB_r_0=linB_r;           rdw=1;}
                if(linB_ri!=linB_ri_0){   linB_ri_0=linB_ri;         rdw=1;}
                if(linB_sr!=linB_sr_0){   linB_sr_0=linB_sr;         rdw=1;}
                if(linB_sri!=linB_sri_0){ linB_sri_0=linB_sri;       rdw=1;}
                if(linB_sR!=linB_sR_0){   linB_sR_0=linB_sR;         rdw=1;}
                if(linB_sRi!=linB_sRi_0){ linB_sRi_0=linB_sRi;       rdw=1;}
                if(linB_sxy!=linB_sxy_0){ linB_sxy_0=linB_sxy;       rdw=1;}
                if(linB_syx!=linB_syx_0){ linB_syx_0=linB_syx;       rdw=1;}
                if(linB_sgxy!=linB_sgxy_0){linB_sgxy_0=linB_sgxy;    rdw=1;}
                if(linB_sgyx!=linB_sgyx_0){linB_sgyx_0=linB_sgyx;    rdw=1;}
                if(fb_r!=fb_r_0){         fb_r_0=fb_r;               rdw=1;}
                if(fb_ri!=fb_ri_0){       fb_ri_0=fb_ri;             rdw=1;}
                if(fb_sr!=fb_sr_0){       fb_sr_0=fb_sr;             rdw=1;}
                if(fb_sri!=fb_sri_0){     fb_sri_0=fb_sri;           rdw=1;}
                if(fb_sR!=fb_sR_0){       fb_sR_0=fb_sR;             rdw=1;}
                if(fb_sRi!=fb_sRi_0){     fb_sRi_0=fb_sRi;           rdw=1;}
                if(fb_sxy!=fb_sxy_0){     fb_sxy_0=fb_sxy;            rdw=1;}
                if(fb_syx!=fb_syx_0){     fb_syx_0=fb_syx;            rdw=1;}
                if(fb_sgxy!=fb_sgxy_0){   fb_sgxy_0=fb_sgxy;          rdw=1;}
                if(fb_sgyx!=fb_sgyx_0){   fb_sgyx_0=fb_sgyx;          rdw=1;}
                if(mod_r!=mod_r_0){       mod_r_0=mod_r;              rdw=1;}
                if(mod_ri!=mod_ri_0){     mod_ri_0=mod_ri;            rdw=1;}
                if(mod_sr!=mod_sr_0){     mod_sr_0=mod_sr;            rdw=1;}
                if(mod_sri!=mod_sri_0){   mod_sri_0=mod_sri;          rdw=1;}
                if(mod_sR!=mod_sR_0){     mod_sR_0=mod_sR;            rdw=1;}
                if(mod_sRi!=mod_sRi_0){   mod_sRi_0=mod_sRi;          rdw=1;}
                if(mod_sxy!=mod_sxy_0){   mod_sxy_0=mod_sxy;          rdw=1;}
                if(mod_syx!=mod_syx_0){   mod_syx_0=mod_syx;          rdw=1;}
                if(mod_sgxy!=mod_sgxy_0){ mod_sgxy_0=mod_sgxy;        rdw=1;}
                if(mod_sgyx!=mod_sgyx_0){ mod_sgyx_0=mod_sgyx;        rdw=1;}
                if(linB_am!=linB_am_0){   linB_am_0=linB_am;          rdw=1;}
                if(linB_sd!=linB_sd_0){   linB_sd_0=linB_sd;          rdw=1;}
                if(linB_sgam!=linB_sgam_0){linB_sgam_0=linB_sgam;     rdw=1;}
                if(linB_sdg!=linB_sdg_0){ linB_sdg_0=linB_sdg;        rdw=1;}
                if(linB_a3!=linB_a3_0){   linB_a3_0=linB_a3;          rdw=1;}
                if(linB_ag3!=linB_ag3_0){ linB_ag3_0=linB_ag3;        rdw=1;}
                if(linB_sga3!=linB_sga3_0){linB_sga3_0=linB_sga3;     rdw=1;}
                if(linB_a4!=linB_a4_0){   linB_a4_0=linB_a4;          rdw=1;}
                if(linB_ag4!=linB_ag4_0){ linB_ag4_0=linB_ag4;        rdw=1;}
                if(linB_sga4!=linB_sga4_0){linB_sga4_0=linB_sga4;     rdw=1;}
                if(linB_e!=linB_e_0){     linB_e_0=linB_e;            rdw=1;}
                if(linB_x!=linB_x_0){     linB_x_0=linB_x;            rdw=1;}
                if(linB_x0!=linB_x0_0){   linB_x0_0=linB_x0;          rdw=1;}
                if(linB_x1!=linB_x1_0){   linB_x1_0=linB_x1;          rdw=1;}
                if(linB_s0!=linB_s0_0){   linB_s0_0=linB_s0;          rdw=1;}
                if(linB_s1!=linB_s1_0){   linB_s1_0=linB_s1;          rdw=1;}
                if(fb_am!=fb_am_0){       fb_am_0=fb_am;              rdw=1;}
                if(fb_sd!=fb_sd_0){       fb_sd_0=fb_sd;              rdw=1;}
                if(fb_sgam!=fb_sgam_0){   fb_sgam_0=fb_sgam;          rdw=1;}
                if(fb_sdg!=fb_sdg_0){     fb_sdg_0=fb_sdg;            rdw=1;}
                if(fb_a3!=fb_a3_0){       fb_a3_0=fb_a3;              rdw=1;}
                if(fb_ag3!=fb_ag3_0){     fb_ag3_0=fb_ag3;            rdw=1;}
                if(fb_sga3!=fb_sga3_0){   fb_sga3_0=fb_sga3;          rdw=1;}
                if(fb_a4!=fb_a4_0){       fb_a4_0=fb_a4;              rdw=1;}
                if(fb_ag4!=fb_ag4_0){     fb_ag4_0=fb_ag4;            rdw=1;}
                if(fb_sga4!=fb_sga4_0){   fb_sga4_0=fb_sga4;          rdw=1;}
                if(fb_e!=fb_e_0){         fb_e_0=fb_e;                rdw=1;}
                if(fb_x!=fb_x_0){         fb_x_0=fb_x;                rdw=1;}
                if(fb_x0!=fb_x0_0){       fb_x0_0=fb_x0;              rdw=1;}
                if(fb_x1!=fb_x1_0){       fb_x1_0=fb_x1;              rdw=1;}
                if(fb_s0!=fb_s0_0){       fb_s0_0=fb_s0;              rdw=1;}
                if(fb_s1!=fb_s1_0){       fb_s1_0=fb_s1;              rdw=1;}
                if(mod_am!=mod_am_0){     mod_am_0=mod_am;            rdw=1;}
                if(mod_sd!=mod_sd_0){     mod_sd_0=mod_sd;            rdw=1;}
                if(mod_sgam!=mod_sgam_0){ mod_sgam_0=mod_sgam;        rdw=1;}
                if(mod_sdg!=mod_sdg_0){   mod_sdg_0=mod_sdg;          rdw=1;}
                if(mod_a3!=mod_a3_0){     mod_a3_0=mod_a3;            rdw=1;}
                if(mod_ag3!=mod_ag3_0){   mod_ag3_0=mod_ag3;          rdw=1;}
                if(mod_sga3!=mod_sga3_0){ mod_sga3_0=mod_sga3;        rdw=1;}
                if(mod_a4!=mod_a4_0){     mod_a4_0=mod_a4;            rdw=1;}
                if(mod_ag4!=mod_ag4_0){   mod_ag4_0=mod_ag4;          rdw=1;}
                if(mod_sga4!=mod_sga4_0){ mod_sga4_0=mod_sga4;        rdw=1;}
                if(mod_e!=mod_e_0){       mod_e_0=mod_e;              rdw=1;}
                if(mod_x!=mod_x_0){       mod_x_0=mod_x;              rdw=1;}
                if(mod_x0!=mod_x0_0){     mod_x0_0=mod_x0;            rdw=1;}
                if(mod_x1!=mod_x1_0){     mod_x1_0=mod_x1;            rdw=1;}
                if(mod_s0!=mod_s0_0){     mod_s0_0=mod_s0;            rdw=1;}
                if(mod_s1!=mod_s1_0){     mod_s1_0=mod_s1;            rdw=1;}
                if(r_q!=r_q_0){           r_q_0=r_q;                  rdw=1;}
                if(ri_q!=ri_q_0){         ri_q_0=ri_q;                rdw=1;}
                if(syx_q !=syx_q_0){      syx_q_0=syx_q ;             rdw=1;}
                if(sxy_q!=sxy_q_0){       sxy_q_0=sxy_q;              rdw=1;}
                if(sgyx_q!=sgyx_q_0){     sgyx_q_0=sgyx_q;            rdw=1;}
                if(sgxy_q !=sgxy_q_0){    sgxy_q_0=sgxy_q ;           rdw=1;}
                if(sr_q !=sr_q_0){        sr_q_0=sr_q ;               rdw=1;}
                if(sri_q !=sri_q_0){      sri_q_0=sri_q ;             rdw=1;}
                if(sR_q!=sR_q_0){         sR_q_0=sR_q;                rdw=1;}
                if(sRi_q!=sRi_q_0){       sRi_q_0=sRi_q;              rdw=1;}
                if(am_q!=am_q_0){         am_q_0=am_q;                rdw=1;}
                if(sd_q !=sd_q_0){        sd_q_0=sd_q ;               rdw=1;}
                if(sdg_q!=sdg_q_0){       sdg_q_0=sdg_q;              rdw=1;}
                if(sgam_q !=sgam_q_0){    sgam_q_0=sgam_q ;           rdw=1;}
                if(a3_q!=a3_q_0){         a3_q_0=a3_q;                rdw=1;}
                if(ag3_q!=ag3_q_0){       ag3_q_0=ag3_q;              rdw=1;}
```

```
               if(sa3g_q !=sa3g_q_0){    sa3g_q_0=sa3g_q ;           rdw=1;}
               if(a4_q!=a4_q_0){         a4_q_0=a4_q;                rdw=1;}
               if(ag4_q!=ag4_q_0){       ag4_q_0=ag4_q;              rdw=1;}
               if(sa4g_q!=sa4g_q_0){     sa4g_q_0=sa4g_q;            rdw=1;}
               if(e_q!=e_q_0){           e_q_0=e_q;                  rdw=1;}
               if(x_q!=x_q_0){           x_q_0=x_q;                  rdw=1;}
               if(x0_q!=x0_q_0){         x0_q_0=x0_q;                rdw=1;}
               if(x1_q!=x1_q_0){         x1_q_0=x1_q;                rdw=1;}
               if(s0_q!=s0_q_0){         s0_q_0=s0_q;                rdw=1;}
               if(s1_q!=s1_q_0){         s1_q_0=s1_q;                rdw=1;}
               if(sw_xb!=sw_xb_0){       sw_xb_0=sw_xb;              rdw=1;}
               if(sw_yb!=sw_yb_0){       sw_yb_0=sw_yb;              rdw=1;}
               if(xBz!=xBz_0){           xBz_0=xBz;                  rdw=1;}
               if(yBz!=yBz_0){           yBz_0=yBz;                  rdw=1;}
               if(p_e_sw!=p_e_sw_0)
               {                         p_e_sw_0=p_e_sw;
                      if(sw_mod_==4){sw_mod_=0;OnModusEpsilon();        rdw=2;}
               }
               if(p_p_sw!=p_p_sw_0)
               {                         p_p_sw_0=p_p_sw;
                      if(sw_mod_==3){sw_mod_=0;OnModusFp();             rdw=2;}
               }
               if(sw_pq!=sw_pq_0)
               {                         sw_pq_0=sw_pq;
                      if(sw_mod_==3){sw_mod_=0;OnModusFp();             rdw=2;}
               }
               if(sw_integral==1){       sw_integral=0;
                                                   integral();              rdw=2;}
               if(sw_integral==2){       sw_integral=0;      fx_0(); rdw=2;}
               if(sw_differential==1){   sw_differential=0;
                                            differential();          rdw=2;}
               if(sw_differential==2){   sw_differential=0; fx_0(); rdw=2;}
               if(sw_kgl==1){            sw_kgl=0;kgl();             rdw=2;}
               if(sw_kgl==2){            sw_kgl=0;           fx_0(); rdw=2;}
               if(sw_fxy==1){            sw_fxy=0;           fx_1(); rdw=2;}


                 if(rdw==1)RedrawWindow();

                       rdw=0;

               if(csr_!=csr_0){csr_0=csr_;OnEinstellungenCursor_();}
       }


       CDialog::OnTimer(nIDEvent);
}

void CGRP2Dlg::OnInfo() {GRP2info o; o.DoModal();}

void CGRP2Dlg::OnEinstellungenKoordinaten() //menue -> dialog Koordinaten einstellung
{
       GRP2Koord o;
                  o.Koord_(mAx,mAy,mVx,mVy, (max_x-min_x)/tlg_x, (max_y-min_y)/tlg_y);
//Koordinatenwertübergabe
                        o.MinMax_(min_x, max_x, min_y, max_y);//kennwertwertübergabe

       sw_mkoord_A=1;
       sw_mkoord_V=1;

                        o.DoModal();
}

void CGRP2Dlg::OnEINSTELLUNGENParameter()  //menue -> dialogparameter einstellung
{
       CWinApp*  pApp = AfxGetApp();

                  pApp->WriteProfileString("Achsen","x_Justierung",ftoc(corx*100,8));
                  pApp->WriteProfileString("Achsen","y_Justierung",ftoc(cory*100,8));

          tab_ = pApp->GetProfileInt("Parameter","Tabelle",1);

       if(tab_==1){ GRP2param1 o;o.DoModal();}
       if(tab_==2){ GRP2param2 o;o.DoModal();}
       if(tab_==3){ GRP2param3 o;o.DoModal();}
       if(tab_==4){ GRP2param4 o;o.DoModal();}
       if(tab_==5){ GRP2param5 o;o.DoModal();}
}
```

```
void CGRP2Dlg::OnEINSTELLUNGENDiagramm() //menue -> dialog diagramm einstellung
{
         CWinApp*  pApp = AfxGetApp();
        tab_diag = pApp->GetProfileInt("Diagramm","Tabelle",1);

        if(tab_diag==1){ GRP2diagrm1 o;o.DoModal();}
        if(tab_diag==2){ GRP2diagrm2 o;o.DoModal();}
        if(tab_diag==3){ GRP2diagrm3 o;o.DoModal();}
        if(tab_diag==4){ GRP2diagrm4 o;o.DoModal();}
        if(tab_diag==5){ GRP2diagrm5 o;o.DoModal();}
        if(tab_diag==6){ GRP2diagrm6 o;o.DoModal();}
}

void CGRP2Dlg::OnEINSTELLUNGENFunktion()//menue -> dialog funktions einstellung
{
        CWinApp*  pApp = AfxGetApp();
         //tab_fn = pApp->GetProfileInt("Funktion","Tabelle",1);

        if(sw_mod_==1){ GRP2fn1 o;o.DoModal();}
        if(sw_mod_==5){ GRP2fn1 o;o.DoModal();}
        if(sw_mod_==2)
        {
               GRP2fn2 o;
                       //o.t_df(n_); über inistream realisiert
                              o.DoModal();
        }
        if(sw_mod_==3){ GRP2fn3 o;o.DoModal();}
        if(sw_mod_==4){ GRP2fn4 o;o.DoModal();}
}

void CGRP2Dlg::OnEinstellungenApp() { GRP2prg o;o.DoModal();}//menue -> dialog programm
einstellung


void CGRP2Dlg::OnFunktionsmatrixFfnen()//menue -> dialog datei öffnen (ascii funktionsmatrix)
{
        filename0=filename;        static char BASED_CODE szFilter[]="ASCII Funktionsmatrix-
Dateien (*.asc) |*.asc|ASCII Text Funktionsmatrixdateien (*.txt) |*.txt|Alle Dateien (*.*)
|*.*||";

         CFileDialog f(1,"asc","*.asc", OFN_HIDEREADONLY |
                                                         OFN_NOVALIDATE  , szFilter);
                     f.DoModal();
       filename= f.GetFileName();

        if(filename!="")
        {
               swli=1; //

               sw_emf_in=0; //

               posX=posY=0; //diagrammverschub nullsetzen
               frmX=frmY=1; //diagrammform ursprung

               file_ind=1;//dateineusetzungsindex

               MINMAX(); fn_THETA_1(); //funktionsparameter

               Fenstertext(filename); filename_tmp=filename;

               if(log_)log_file(1); //Funktionsmatrix log;
               if(sw_Log){sw_Log=0;OnAnsichtGrp2log(1);} //Log Ansicht aktualisieren

               sw_mod_=-1; OnModusFx();
        }

        if(filename=="")  filename=filename0;
}

void CGRP2Dlg::OnFunktionEmfffnen() //menue -> dialog emf datei öffnen
{
        emf_filename0=emf_filename;
                                   static char BASED_CODE szFilter[]="EMF Grafik-Dateien
(*.emf) |*.emf||";

        CFileDialog  f(1,"emf","*.emf", OFN_HIDEREADONLY |
                                                         OFN_NOVALIDATE  , szFilter);
```

```
                    f.DoModal();
       emf_filename= f.GetFileName();

        if(emf_filename!="")
        {
               sw_emf_in=1;

               Fenstertext(emf_filename);

               if(sw_Theta){sw_Theta=1;OnAnsichtThetafensterq(0);} //Theta Ansicht schliessen
               if(sw_Fxy){sw_Fxy=1;OnAnsichtFunktionsmatrixfxy(0);} //Fxy Ansicht schliessen

               OnEinstellungenCursorSchwarz();

                 CMenu o;
                              o.LoadMenu(IDR_MENU2); //emf menue
               SetMenu(&o);

               RedrawWindow();
        }

        if(emf_filename=="")  emf_filename=emf_filename0;
}

void CGRP2Dlg::Fenstertext(CString fText) //Hauptfenstertext
{
          CString c;
                         c= "GRP2 - [";
                            c+= fText;
                                c+= "]";
        SetWindowText(c);
}

void CGRP2Dlg::OnFunktionEmfspeichern() //menue ->  emf datei speichern
{
        //emfname="GRP_Diagramm_";
        CWinApp* pApp = AfxGetApp();

        emfname= pApp->GetProfileString("EMF","Dateiname","GRP_Diagramm_");

        emfname+=itoc(time(0));
        emfname+=".emf";

        sw_emf=1; //emf schalter ein

        RedrawWindow();        //emf graphik unter 'emfname' generieren
}

void CGRP2Dlg::OnFunktionEmfspeichernunter() //menue ->  emf datei speichern unter
{
        int i=0;

        CString file_ext, file_;        static char BASED_CODE szFilter[]="EMF Grafik-Dateien
(*.emf)  |*.emf||";

        CFileDialog  f(0,"emf","*.emf", OFN_HIDEREADONLY    |
                                                        OFN_NOVALIDATE      |
                                                                  OFN_NOCHANGEDIR     | //
                                                                  OFN_OVERWRITEPROMPT |
                                    OFN_SHAREAWARE      |
                                                                  OFN_PATHMUSTEXIST   ,
szFilter);
                    f.DoModal();
                 file_= f.GetPathName();
          file_ext= f.GetFileExt();

        if(file_ext!="")if(file_!="")if(file_.Right(5)!="*.emf") //ggf erweitern...
        {
               emfname=file_;

               sw_emf=1; //emf schalter ein

               RedrawWindow();        //emf graphik unter 'emfname' generieren
        }
}


void CGRP2Dlg::OnFunktionSpeichern() //menue ->  speichern
```

```
{
	FILE  *outStream;

	int i=0;

	outStream = fopen(filename_tmp, "wb" );

	do //funktionsmatrix in datei schreiben
	{
		if(i>0)
		fprintf(outStream,"\n");
		fprintf(outStream,"%f\t%f",FVx_[i],FVy_[i]);

			i++;
	}while (i<n_);

	fclose( outStream );

	if(log_)log_file(2); //Funktionsmatrix log;
	if(sw_Log){sw_Log=0;OnAnsichtGrp2log(1);} //Log Ansicht aktualisieren
}


void CGRP2Dlg::OnFunktionSpeichernunter() //menue ->  speichern unter
{
	FILE  *outStream;

	int i=0;

	CString file_ext, file_;	static char BASED_CODE szFilter[]="ASCII 
Funktionsmatrix-Dateien (*.asc) |*.asc|ASCII Text Funktionsmatrixdateien (*.txt) |*.txt|Alle 
Dateien (*.*) |*.*||";

	CFileDialog  f(0,"asc","*.asc", OFN_HIDEREADONLY    |
						     OFN_NOVALIDATE      |
							OFN_NOCHANGEDIR     | //
							OFN_OVERWRITEPROMPT |
				     OFN_SHAREAWARE      |
							OFN_PATHMUSTEXIST   ,
szFilter);
		    f.DoModal();
		 file_= f.GetPathName();
	   file_ext= f.GetFileExt();

	if(file_ext!="")if(file_!="")if(fopen(file_, "wb" )!=0)
	{
		outStream = fopen(file_, "wb" );

		do //funktionsmatrix in datei schreiben
		{
			if(i>0)
			fprintf(outStream,"\n");
			fprintf(outStream,"%f\t%f",FVx_[i],FVy_[i]);

				i++;
		}while (i<n_);

		fclose( outStream );
	}
}


void CGRP2Dlg::OnAnsichtFunktionsmatrixfxy(UINT sw) //menue -> ansicht (Fxy)
{
	sw_Fxy=!sw_Fxy;
				 CWinApp*  pApp = AfxGetApp();

	if(sw_Fxy)
	{
		if(sw!=1)m_FxyDlg.Create(IDD_GRP2_FXY); //Funktionsmatrixfenster create bei 
nicht aktualisierung
		if(sw!=1)m_FxyDlg.ShowWindow(SW_SHOW);

		m_FxyDlg.SetWindowPos(&wndTop, pApp->GetProfileInt("Fxy","Position_x",100),
					      pApp->GetProfileInt("Fxy","Position_y",100),
0,0, SWP_NOSIZE);//Fxyfensterposition

		CString stxt;
```

```
               CString cfxy;
                       cfxy="";

                  int i=0;

               do  //Funktionsmatrix einlesen
               {
                                                            stxt= itoc(i+1); //zähler
                                     m_FxyDlg.SetWindowText(stxt);

                                                                                 cfxy+=stxt;
                                                                                 cfxy+=":";
                                                                                 cfxy+="\t";

       cfxy+=ftoc(FVx_[i],5); //Funktionsvektor (x)
                                                                                 cfxy+="   ";

       cfxy+=ftoc(FVy_[i],5); //Funktionsvektor (y)

       cfxy+="\x0d\x0a";      //Absatz
                         i++;
               }while(i<n_);

               m_FxyDlg.SetDlgItemText(IDC_EDIT_FXY,cfxy);//funktionsmatrix darstellen

                                      stxt= "(F)[";
            if(!sw_inv)
                      {
                 if(sw_mod_==1)stxt+="f(x)";
                               if(sw_mod_==5)stxt+="f(zx)";
                               if(sw_mod_==2)stxt+="r(x,y)";
                               if(sw_mod_==3)stxt+="F(p)";
                               if(sw_mod_==4)stxt+="F(e)";
                      }

                      if( sw_inv)
                      {
                 if(sw_mod_==1)stxt+="f(y)";
                               if(sw_mod_==5)stxt+="f(zy)";
                               if(sw_mod_==2)stxt+="r(y,x)";
                               if(sw_mod_==3)stxt+="F(q)";
                               if(sw_mod_==4)stxt+="F(p)";
                      }
                                                       stxt+="] n=";
                                                       stxt+= itoc(n_);
               m_FxyDlg.SetWindowText(stxt);           //fenstertext
       }

       if(!sw_Fxy)
       {
                                                  CRect coordfxy; //lokale struktur
                                          m_FxyDlg.GetWindowRect(&coordfxy);

               pApp->WriteProfileInt("Fxy","Position_x",coordfxy.left);//Fxyfensterposition
speichern
               pApp->WriteProfileInt("Fxy","Position_y",coordfxy.top); //

               m_FxyDlg.DestroyWindow();
       }

                 CMenu o;
                                     o.LoadMenu(IDR_MENU1);
       if(sw_inv==1)
       {
                                     o.ModifyMenu(ID_MODUS_FX,MF_BYCOMMAND,ID_MODUS_FX,"f-
1(x)=f(y)");
                                     o.ModifyMenu(ID_MODUS_FZX,MF_BYCOMMAND,ID_MODUS_FZX,"f-
1(zx)=f(zy)");

       o.ModifyMenu(ID_MODUS_RXY,MF_BYCOMMAND,ID_MODUS_RXY,"r(y,x)");
                                     o.ModifyMenu(ID_MODUS_FP,MF_BYCOMMAND,ID_MODUS_FP,"F-
1(p)=F(q)");

       o.ModifyMenu(ID_MODUS_EPSILON,MF_BYCOMMAND,ID_MODUS_EPSILON,"F-1(e)=F(p)");
                                     o.CheckMenuItem(ID_MODUS_INV,  MF_CHECKED);
       }
       if(sw_mod_==1)  o.CheckMenuItem(ID_MODUS_FX,  MF_CHECKED);
       if(sw_mod_==2)  o.CheckMenuItem(ID_MODUS_RXY,  MF_CHECKED);
       if(sw_mod_==3)  o.CheckMenuItem(ID_MODUS_FP,  MF_CHECKED);
       if(sw_mod_==4)  o.CheckMenuItem(ID_MODUS_EPSILON,  MF_CHECKED);
       if(sw_mod_==5)  o.CheckMenuItem(ID_MODUS_FZX,  MF_CHECKED);
       if(sw_csr==1)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_WEISS,  MF_CHECKED);
       if(sw_csr==2)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_SCHWARZ,  MF_CHECKED);
       if(sw_As==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_ACHSENVERSCHUB, MF_CHECKED);
       if(sw_Vs==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_VEKTORENVERSCHUB,  MF_CHECKED);
       if(sw_Fxy==1)   o.CheckMenuItem(ID_ANSICHT_FUNKTIONSMATRIXFXY, MF_CHECKED);
       if(sw_Theta==1) o.CheckMenuItem(ID_ANSICHT_THETAFENSTERQ, MF_CHECKED);
       if(sw_Log==1)   o.CheckMenuItem(ID_ANSICHT_GRP2LOG, MF_CHECKED);
       if(sw_status==1)o.CheckMenuItem(ID_ANSICHT_STATUSLEISTE, MF_CHECKED);


              SetMenu(&o);
}

void CGRP2Dlg::OnAnsichtThetafensterq(UINT sw) //menue -> ansicht Theta
{
       sw_Theta=!sw_Theta;
                                   CWinApp*  pApp = AfxGetApp();

       if(sw_Theta)
       {
              if(sw!=1)m_ThetaDlg.Create(IDD_GRP2_THETA); //Thetafenster create bei nicht
aktualisierung
              if(sw!=1)m_ThetaDlg.ShowWindow(SW_SHOW);

              m_ThetaDlg.SetWindowPos(&wndTop, pApp->GetProfileInt("Theta","Position_x",100),
                                 pApp->GetProfileInt("Theta","Position_y",100),
0,0, SWP_NOSIZE);//Thetafensterposition

              CString stxt;

              CString cfxy;
                     cfxy="";

           //Thetamatrix generieren

              if(sw_mod_==1) //fx
              {
                     cfxy+="n:";  cfxy+="\t";cfxy+=itoc(n_);     cfxy+="\t";cfxy+=itoc(n_);
cfxy+="\x0d\x0a";

       cfxy+="min:";cfxy+="\t";cfxy+=ftoc(min_x,3);cfxy+="\t";cfxy+=ftoc(min_y,3);cfxy+="\x0d\
x0a";

       cfxy+="max:";cfxy+="\t";cfxy+=ftoc(max_x,3);cfxy+="\t";cfxy+=ftoc(max_y,3);cfxy+="\x0d\
x0a";
                     cfxy+="R:";cfxy+="\t";cfxy+=ftoc(max_x-
min_x,3);cfxy+="\t";cfxy+=ftoc(max_y-min_y,3);cfxy+="\x0d\x0a";
                     if(!sw_inv)
                     {

       cfxy+="am:";cfxy+="\t";cfxy+=ftoc(qX.am,3);cfxy+="\t";cfxy+=ftoc(qY.am,3);cfxy+="\x0d\x
0a";

       cfxy+="s:";cfxy+="\t";cfxy+=ftoc(qX.sd,3);cfxy+="\t";cfxy+=ftoc(qY.sd,3);cfxy+="\x0d\x0
a";

       cfxy+="s²:";cfxy+="\t";cfxy+=ftoc(pow(qX.sd,2),3);cfxy+="\t";cfxy+=ftoc(pow(qY.sd,2),3)
;cfxy+="\x0d\x0a";
                     }

                     if(sw_inv)
                     {

       cfxy+="am:";cfxy+="\t";cfxy+=ftoc(qY.am,3);cfxy+="\t";cfxy+=ftoc(qX.am,3);cfxy+="\x0d\x
0a";

       cfxy+="s:";cfxy+="\t";cfxy+=ftoc(qY.sd,3);cfxy+="\t";cfxy+=ftoc(qX.sd,3);cfxy+="\x0d\x0
a";

       cfxy+="s²:";cfxy+="\t";cfxy+=ftoc(pow(qY.sd,2),3);cfxy+="\t";cfxy+=ftoc(pow(qX.sd,2),3)
;cfxy+="\x0d\x0a";
                     }
              }
```

```
                if(sw_mod_==2) //rxy
                {
                        cfxy+="n:";  cfxy+="\t";cfxy+=itoc(n_);   cfxy+="\x0d\x0a";
                        cfxy+="r:";  cfxy+="\t";cfxy+=ftoc(qR,3); cfxy+="\x0d\x0a";
                        cfxy+="r²:"; cfxy+="\t";cfxy+=ftoc(pow(qR,2),3); cfxy+="\x0d\x0a";
                        cfxy+="s²xy:"; cfxy+="\t";cfxy+=ftoc(qR*qX.sd*qY.sd,3);
cfxy+="\x0d\x0a";
                        cfxy+="ô²xy:"; cfxy+="\t";cfxy+=ftoc((qR*qX.sd*qY.sd)*(n_/(n_-1)),3);
cfxy+="\x0d\x0a";
                        cfxy+="sq':";cfxy+="\t";cfxy+=ftoc(qS,3); cfxy+="\x0d\x0a";
                        cfxy+="ôq':";cfxy+="\t";cfxy+=ftoc(qsS,3);cfxy+="\x0d\x0a";
                        cfxy+="ôr:"; cfxy+="\t";cfxy+=ftoc(qsR,3);cfxy+="\x0d\x0a";
                }

                if(sw_mod_==3) //Fp
                {
                        cfxy+="n:";  cfxy+="\t";cfxy+=itoc(n_);cfxy+="\x0d\x0a";
                        cfxy+="am:"; cfxy+="\t";cfxy+=ftoc(qY.am,3); cfxy+="\x0d\x0a";
                        cfxy+="s:";  cfxy+="\t";cfxy+=ftoc(qY.sd,3); cfxy+="\x0d\x0a";
                        cfxy+="ô:";  cfxy+="\t";cfxy+=ftoc(qY.sdg,3);cfxy+="\x0d\x0a";
                        cfxy+="ôam:";cfxy+="\t";cfxy+=ftoc(qY.sgam,3);cfxy+="\x0d\x0a";
                        cfxy+="a3:"; cfxy+="\t";cfxy+=ftoc(qY.a3,3);cfxy+="\x0d\x0a";
                        cfxy+="â3:"; cfxy+="\t";cfxy+=ftoc(qY.ag3,3);cfxy+="\x0d\x0a";
                        cfxy+="ôa3:";cfxy+="\t";cfxy+=ftoc(qY.sga3,3);cfxy+="\x0d\x0a";
                        cfxy+="a4:"; cfxy+="\t";cfxy+=ftoc(qY.a4,3);cfxy+="\x0d\x0a";
                        cfxy+="â4:"; cfxy+="\t";cfxy+=ftoc(qY.ag4,3);cfxy+="\x0d\x0a";
                        cfxy+="ôa4:";cfxy+="\t";cfxy+=ftoc(qY.sga4,3);cfxy+="\x0d\x0a";
                }

                if(sw_mod_==4) //Fe
                {

                        cfxy+="n:";     cfxy+="\t";cfxy+=itoc(n_/2);
cfxy+="\t";cfxy+=itoc(n_/2);   cfxy+="\x0d\x0a";
                        cfxy+="e:";     cfxy+="\t";cfxy+=ftoc(qE.e,3);
cfxy+="\t";cfxy+=ftoc(qE.e,3); cfxy+="\x0d\x0a";
                        cfxy+="xcrit:"; cfxy+="\t";cfxy+=ftoc(qE.sw,3);
cfxy+="\t";cfxy+=ftoc(qE.sw,3);cfxy+="\x0d\x0a";

        cfxy+="am:";cfxy+="\t";cfxy+=ftoc(qX.am,3);cfxy+="\t";cfxy+=ftoc(qY.am,3);cfxy+="\x0d\x
0a";
                        cfxy+="s:";  cfxy+="\t";cfxy+=ftoc(qX.sd,3);
cfxy+="\t";cfxy+=ftoc(qY.sd,3);cfxy+="\x0d\x0a";
                        cfxy+="ô:";
cfxy+="\t";cfxy+=ftoc(qX.sdg,3);cfxy+="\t";cfxy+=ftoc(qY.sdg,3);cfxy+="\x0d\x0a";
                }

                if(sw_mod_==5) //fzx
                {
                        cfxy+="n:";  cfxy+="\t";cfxy+=itoc(n_);     cfxy+="\t";cfxy+=itoc(n_);
cfxy+="\x0d\x0a";

        cfxy+="min:";cfxy+="\t";cfxy+=ftoc(min_x,3);cfxy+="\t";cfxy+=ftoc(min_y,3);cfxy+="\x0d\
x0a";

        cfxy+="max:";cfxy+="\t";cfxy+=ftoc(max_x,3);cfxy+="\t";cfxy+=ftoc(max_y,3);cfxy+="\x0d\
x0a";
                        cfxy+="R:";cfxy+="\t";cfxy+=ftoc(max_x-
min_x,3);cfxy+="\t";cfxy+=ftoc(max_y-min_y,3);cfxy+="\x0d\x0a";
                        if(!sw_inv)
                        {

        cfxy+="am:";cfxy+="\t";cfxy+="0.000";cfxy+="\t";cfxy+="0.000";cfxy+="\x0d\x0a";

        cfxy+="s:";cfxy+="\t";cfxy+=ftoc(qX.sd,3);cfxy+="\t";cfxy+=ftoc(qY.sd,3);cfxy+="\x0d\x0
a";

        cfxy+="s²:";cfxy+="\t";cfxy+=ftoc(pow(qX.sd,2),3);cfxy+="\t";cfxy+=ftoc(pow(qY.sd,2),3)
;cfxy+="\x0d\x0a";
                        }

                        if(sw_inv)
                        {

        cfxy+="am:";cfxy+="\t";cfxy+="0.000";cfxy+="\t";cfxy+="0.000";cfxy+="\x0d\x0a";

        cfxy+="s:";cfxy+="\t";cfxy+=ftoc(qY.sd,3);cfxy+="\t";cfxy+=ftoc(qX.sd,3);cfxy+="\x0d\x0
a";
```

```
       cfxy+="s²:";cfxy+="\t";cfxy+=ftoc(pow(qY.sd,2),3);cfxy+="\t";cfxy+=ftoc(pow(qX.sd,2),3)
;cfxy+="\x0d\x0a";
                      }
               }

              m_ThetaDlg.SetDlgItemText(IDC_EDIT_THETA,cfxy);//thetamatrix darstellen

                                     stxt= "(Q)[";
                      if(!sw_inv)
                      {
                 if(sw_mod_==1)stxt+="f(x)";
                              if(sw_mod_==5)stxt+="f(zx)";
                              if(sw_mod_==2)stxt+="r(x,y)";
                              if(sw_mod_==3)stxt+="F(p)";
                              if(sw_mod_==4)stxt+="F(e)";
                      }

                      if( sw_inv)
                      {
                 if(sw_mod_==1)stxt+="f(y)";
                              if(sw_mod_==5)stxt+="f(zy)";
                              if(sw_mod_==2)stxt+="r(y,x)";
                              if(sw_mod_==3)stxt+="F(q)";
                              if(sw_mod_==4)stxt+="F(p)";
                      }

                                                      stxt+="]";
              m_ThetaDlg.SetWindowText(stxt);            //fenstertext

       }

       if(!sw_Theta)
       {
                                                                              CRect coordq;
//lokale struktur
                                          m_ThetaDlg.GetWindowRect(&coordq);

              pApp->WriteProfileInt("Theta","Position_x",coordq.left);//Thetafensterposition
speichern
              pApp->WriteProfileInt("Theta","Position_y",coordq.top); //

              m_ThetaDlg.DestroyWindow();
       }

       CMenu o;
                                     o.LoadMenu(IDR_MENU1);
       if(sw_inv==1)
       {
                                     o.ModifyMenu(ID_MODUS_FX,MF_BYCOMMAND,ID_MODUS_FX,"f-
1(x)=f(y)");
                                     o.ModifyMenu(ID_MODUS_FZX,MF_BYCOMMAND,ID_MODUS_FZX,"f-
1(zx)=f(zy)");

       o.ModifyMenu(ID_MODUS_RXY,MF_BYCOMMAND,ID_MODUS_RXY,"r(y,x)");
                                     o.ModifyMenu(ID_MODUS_FP,MF_BYCOMMAND,ID_MODUS_FP,"F-
1(p)=F(q)");

       o.ModifyMenu(ID_MODUS_EPSILON,MF_BYCOMMAND,ID_MODUS_EPSILON,"F-1(e)=F(p)");
                                     o.CheckMenuItem(ID_MODUS_INV,  MF_CHECKED);
       }
       if(sw_mod_==1)  o.CheckMenuItem(ID_MODUS_FX,  MF_CHECKED);
       if(sw_mod_==2)  o.CheckMenuItem(ID_MODUS_RXY,  MF_CHECKED);
       if(sw_mod_==3)  o.CheckMenuItem(ID_MODUS_FP,  MF_CHECKED);
       if(sw_mod_==4)  o.CheckMenuItem(ID_MODUS_EPSILON,  MF_CHECKED);
       if(sw_mod_==5)  o.CheckMenuItem(ID_MODUS_FZX,  MF_CHECKED);
       if(sw_csr==1)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_WEISS,  MF_CHECKED);
       if(sw_csr==2)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_SCHWARZ,  MF_CHECKED);
       if(sw_As==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_ACHSENVERSCHUB, MF_CHECKED);
       if(sw_Vs==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_VEKTORENVERSCHUB,  MF_CHECKED);
       if(sw_Fxy==1)   o.CheckMenuItem(ID_ANSICHT_FUNKTIONSMATRIXFXY, MF_CHECKED);
       if(sw_Theta==1) o.CheckMenuItem(ID_ANSICHT_THETAFENSTERQ, MF_CHECKED);
       if(sw_Log==1)   o.CheckMenuItem(ID_ANSICHT_GRP2LOG, MF_CHECKED);
       if(sw_status==1)o.CheckMenuItem(ID_ANSICHT_STATUSLEISTE, MF_CHECKED);

              SetMenu(&o);
}
```

```
void CGRP2Dlg::OnAnsichtGrp2log(UINT sw) //menue -> ansicht Log
{
	sw_Log=!sw_Log;

					CWinApp*  pApp = AfxGetApp();
	if(sw_Log)
	{
		if(sw!=1)m_LogDlg.Create(IDD_GRP2_LOG); //Funktionsmatrixfenster create bei
nicht aktualisierung
		if(sw!=1)m_LogDlg.ShowWindow(SW_SHOW);

		m_LogDlg.SetWindowPos(&wndTop, pApp->GetProfileInt("Log","Position_x",100),
						pApp->GetProfileInt("Log","Position_y",100),
0,0, SWP_NOSIZE);//logfensterposition

		CString stxt;
			stxt="[Log] n=";
				stxt+= itoc(nlog-1);

		m_LogDlg.SetWindowText(stxt);

		//Log File einlesen

		char log_str[10000][100];
		int i=0;

			    FILE *f;
				f = fopen (log_filename, "r");

		CString c_;			CString clog;
			   /* c_= fgetc(f); */		   clog="";
		do
		{
			 /*if(c_=='\n')			clog+="\x0d\x0a"; //
		     else			      clog+=c_;//

			c_= fgetc(f);*/

		       fgets(log_str[i],100,f);
				      i++;

		}while (    feof (f) == 0);
			   fclose( f );


		if(1)for(int iLauf=0;iLauf<=i-2;iLauf++) {clog+=ctoc(log_str[i-2-
iLauf]);clog+="\x0d\x0a";} //revers


		m_LogDlg.SetDlgItemText(IDC_EDIT_LOG,clog);//logdatei darstellen

	}

	if(!sw_Log)
	{
										CRect
coordlog;//lokale struktur

m_LogDlg.GetWindowRect(&coordlog);

						pApp-
>WriteProfileInt("Log","Position_x",coordlog.left);//logfensterposition speichern
						pApp-
>WriteProfileInt("Log","Position_y",coordlog.top); //
		m_LogDlg.DestroyWindow();
	}

		  CMenu o;
					o.LoadMenu(IDR_MENU1);
	if(sw_inv==1)
	{
					o.ModifyMenu(ID_MODUS_FX,MF_BYCOMMAND,ID_MODUS_FX,"f-
1(x)=f(y)");
					o.ModifyMenu(ID_MODUS_FZX,MF_BYCOMMAND,ID_MODUS_FZX,"f-
1(zx)=f(zy)");

	o.ModifyMenu(ID_MODUS_RXY,MF_BYCOMMAND,ID_MODUS_RXY,"r(y,x)");
```

```
                                     o.ModifyMenu(ID_MODUS_FP,MF_BYCOMMAND,ID_MODUS_FP,"F-
1(p)=F(q)");

        o.ModifyMenu(ID_MODUS_EPSILON,MF_BYCOMMAND,ID_MODUS_EPSILON,"F-1(e)=F(p)");
                                     o.CheckMenuItem(ID_MODUS_INV,  MF_CHECKED);
        }
        if(sw_mod_==1)  o.CheckMenuItem(ID_MODUS_FX,  MF_CHECKED);
        if(sw_mod_==2)  o.CheckMenuItem(ID_MODUS_RXY,  MF_CHECKED);
        if(sw_mod_==3)  o.CheckMenuItem(ID_MODUS_FP,  MF_CHECKED);
        if(sw_mod_==4)  o.CheckMenuItem(ID_MODUS_EPSILON,  MF_CHECKED);
        if(sw_mod_==5)  o.CheckMenuItem(ID_MODUS_FZX,  MF_CHECKED);
        if(sw_csr==1)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_WEISS,  MF_CHECKED);
        if(sw_csr==2)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_SCHWARZ,  MF_CHECKED);
        if(sw_As==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_ACHSENVERSCHUB, MF_CHECKED);
        if(sw_Vs==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_VEKTORENVERSCHUB,  MF_CHECKED);
        if(sw_Fxy==1)   o.CheckMenuItem(ID_ANSICHT_FUNKTIONSMATRIXFXY, MF_CHECKED);
        if(sw_Theta==1) o.CheckMenuItem(ID_ANSICHT_THETAFENSTERQ, MF_CHECKED);
        if(sw_Log==1)   o.CheckMenuItem(ID_ANSICHT_GRP2LOG, MF_CHECKED);
        if(sw_status==1)o.CheckMenuItem(ID_ANSICHT_STATUSLEISTE, MF_CHECKED);

               SetMenu(&o);
}

void CGRP2Dlg::OnAnsichtStatusleiste()
{
        sw_status=!sw_status;

                        CMenu o;
                                     o.LoadMenu(IDR_MENU1);
        if(sw_inv==1)
        {
                                     o.ModifyMenu(ID_MODUS_FX,MF_BYCOMMAND,ID_MODUS_FX,"f-
1(x)=f(y)");
                                     o.ModifyMenu(ID_MODUS_FZX,MF_BYCOMMAND,ID_MODUS_FZX,"f-
1(zx)=f(zy)");

        o.ModifyMenu(ID_MODUS_RXY,MF_BYCOMMAND,ID_MODUS_RXY,"r(y,x)");
                                     o.ModifyMenu(ID_MODUS_FP,MF_BYCOMMAND,ID_MODUS_FP,"F-
1(p)=F(q)");

        o.ModifyMenu(ID_MODUS_EPSILON,MF_BYCOMMAND,ID_MODUS_EPSILON,"F-1(e)=F(p)");
                                     o.CheckMenuItem(ID_MODUS_INV,  MF_CHECKED);
        }
        if(sw_mod_==1)  o.CheckMenuItem(ID_MODUS_FX,  MF_CHECKED);
        if(sw_mod_==2)  o.CheckMenuItem(ID_MODUS_RXY,  MF_CHECKED);
        if(sw_mod_==3)  o.CheckMenuItem(ID_MODUS_FP,  MF_CHECKED);
        if(sw_mod_==4)  o.CheckMenuItem(ID_MODUS_EPSILON,  MF_CHECKED);
        if(sw_mod_==5)  o.CheckMenuItem(ID_MODUS_FZX,  MF_CHECKED);
        if(sw_csr==1)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_WEISS,  MF_CHECKED);
        if(sw_csr==2)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_SCHWARZ,  MF_CHECKED);
        if(sw_As==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_ACHSENVERSCHUB, MF_CHECKED);
        if(sw_Vs==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_VEKTORENVERSCHUB,  MF_CHECKED);
        if(sw_Fxy==1)   o.CheckMenuItem(ID_ANSICHT_FUNKTIONSMATRIXFXY, MF_CHECKED);
        if(sw_Theta==1) o.CheckMenuItem(ID_ANSICHT_THETAFENSTERQ, MF_CHECKED);
        if(sw_Log==1)   o.CheckMenuItem(ID_ANSICHT_GRP2LOG, MF_CHECKED);
        if(sw_status==1)o.CheckMenuItem(ID_ANSICHT_STATUSLEISTE, MF_CHECKED);

               SetMenu(&o);

        RedrawWindow();
}



void CGRP2Dlg::OnEinstellungenCursorAchsenverschub()
{
        sw_As=!sw_As;//achsenshiftschalter

                     CMenu o;
                                     o.LoadMenu(IDR_MENU1);
        if(sw_inv==1)
        {
                                     o.ModifyMenu(ID_MODUS_FX,MF_BYCOMMAND,ID_MODUS_FX,"f-
1(x)=f(y)");
                                     o.ModifyMenu(ID_MODUS_FZX,MF_BYCOMMAND,ID_MODUS_FZX,"f-
1(zx)=f(zy)");

        o.ModifyMenu(ID_MODUS_RXY,MF_BYCOMMAND,ID_MODUS_RXY,"r(y,x)");
```

```
                                     o.ModifyMenu(ID_MODUS_FP,MF_BYCOMMAND,ID_MODUS_FP,"F-
1(p)=F(q)");

        o.ModifyMenu(ID_MODUS_EPSILON,MF_BYCOMMAND,ID_MODUS_EPSILON,"F-1(e)=F(p)");
                                     o.CheckMenuItem(ID_MODUS_INV,  MF_CHECKED);
        }
        if(sw_mod_==1)  o.CheckMenuItem(ID_MODUS_FX,  MF_CHECKED);
        if(sw_mod_==2)  o.CheckMenuItem(ID_MODUS_RXY,  MF_CHECKED);
        if(sw_mod_==3)  o.CheckMenuItem(ID_MODUS_FP,  MF_CHECKED);
        if(sw_mod_==4)  o.CheckMenuItem(ID_MODUS_EPSILON,  MF_CHECKED);
        if(sw_mod_==5)  o.CheckMenuItem(ID_MODUS_FZX,  MF_CHECKED);
        if(sw_csr==1)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_WEISS,  MF_CHECKED);
        if(sw_csr==2)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_SCHWARZ,  MF_CHECKED);
        if(sw_As==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_ACHSENVERSCHUB, MF_CHECKED);
        if(sw_Vs==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_VEKTORENVERSCHUB,  MF_CHECKED);
        if(sw_Fxy==1)   o.CheckMenuItem(ID_ANSICHT_FUNKTIONSMATRIXFXY, MF_CHECKED);
        if(sw_Theta==1) o.CheckMenuItem(ID_ANSICHT_THETAFENSTERQ, MF_CHECKED);
        if(sw_Log==1)   o.CheckMenuItem(ID_ANSICHT_GRP2LOG, MF_CHECKED);
        if(sw_status==1)o.CheckMenuItem(ID_ANSICHT_STATUSLEISTE, MF_CHECKED);

            SetMenu(&o);
}

void CGRP2Dlg::OnEinstellungenCursorVektorenverschub()
{
        sw_Vs=!sw_Vs;//vektorshiftschalter

        CMenu o;
                                     o.LoadMenu(IDR_MENU1);
        if(sw_inv==1)
        {
                                     o.ModifyMenu(ID_MODUS_FX,MF_BYCOMMAND,ID_MODUS_FX,"f-
1(x)=f(y)");
                                     o.ModifyMenu(ID_MODUS_FZX,MF_BYCOMMAND,ID_MODUS_FZX,"f-
1(zx)=f(zy)");

        o.ModifyMenu(ID_MODUS_RXY,MF_BYCOMMAND,ID_MODUS_RXY,"r(y,x)");
                                     o.ModifyMenu(ID_MODUS_FP,MF_BYCOMMAND,ID_MODUS_FP,"F-
1(p)=F(q)");

        o.ModifyMenu(ID_MODUS_EPSILON,MF_BYCOMMAND,ID_MODUS_EPSILON,"F-1(e)=F(p)");
                                     o.CheckMenuItem(ID_MODUS_INV,  MF_CHECKED);
        }
        if(sw_mod_==1)  o.CheckMenuItem(ID_MODUS_FX,  MF_CHECKED);
        if(sw_mod_==2)  o.CheckMenuItem(ID_MODUS_RXY,  MF_CHECKED);
        if(sw_mod_==3)  o.CheckMenuItem(ID_MODUS_FP,  MF_CHECKED);
        if(sw_mod_==4)  o.CheckMenuItem(ID_MODUS_EPSILON,  MF_CHECKED);
        if(sw_mod_==5)  o.CheckMenuItem(ID_MODUS_FZX,  MF_CHECKED);
        if(sw_csr==1)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_WEISS,  MF_CHECKED);
        if(sw_csr==2)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_SCHWARZ,  MF_CHECKED);
        if(sw_As==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_ACHSENVERSCHUB, MF_CHECKED);
        if(sw_Vs==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_VEKTORENVERSCHUB, MF_CHECKED);
        if(sw_Fxy==1)   o.CheckMenuItem(ID_ANSICHT_FUNKTIONSMATRIXFXY, MF_CHECKED);
        if(sw_Theta==1) o.CheckMenuItem(ID_ANSICHT_THETAFENSTERQ, MF_CHECKED);
        if(sw_Log==1)   o.CheckMenuItem(ID_ANSICHT_GRP2LOG, MF_CHECKED);
        if(sw_status==1)o.CheckMenuItem(ID_ANSICHT_STATUSLEISTE, MF_CHECKED);

              SetMenu(&o);
}


void CGRP2Dlg::OnModusFx() //f(x)
{
        if(sw_mod_!=1)
        {
                                CMenu o;
                                             o.LoadMenu(IDR_MENU1);
               if(sw_inv==1)
               {

        o.ModifyMenu(ID_MODUS_FX,MF_BYCOMMAND,ID_MODUS_FX,"f-1(x)=f(y)");

        o.ModifyMenu(ID_MODUS_FZX,MF_BYCOMMAND,ID_MODUS_FZX,"f-1(zx)=f(zy)");

        o.ModifyMenu(ID_MODUS_RXY,MF_BYCOMMAND,ID_MODUS_RXY,"r(y,x)");

        o.ModifyMenu(ID_MODUS_FP,MF_BYCOMMAND,ID_MODUS_FP,"F-1(p)=F(q)");
        o.ModifyMenu(ID_MODUS_EPSILON,MF_BYCOMMAND,ID_MODUS_EPSILON,"F-1(e)=F(p)");
                                              o.CheckMenuItem(ID_MODUS_INV,  MF_CHECKED);
                }
                                              o.CheckMenuItem(ID_MODUS_FX,  MF_CHECKED);      //
                                              o.CheckMenuItem(ID_MODUS_FZX,  MF_UNCHECKED);
                                              o.CheckMenuItem(ID_MODUS_FP,  MF_UNCHECKED);
                                              o.CheckMenuItem(ID_MODUS_RXY, MF_UNCHECKED);
                                              o.CheckMenuItem(ID_MODUS_EPSILON,  MF_UNCHECKED);
                if(sw_inv==0)   o.CheckMenuItem(ID_MODUS_INV,  MF_UNCHECKED);
                if(sw_inv==1)   o.CheckMenuItem(ID_MODUS_INV,  MF_CHECKED);
                if(sw_csr==1)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_WEISS,  MF_CHECKED);
                if(sw_csr==2)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_SCHWARZ,  MF_CHECKED);
                if(sw_As==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_ACHSENVERSCHUB,
MF_CHECKED);
            if(sw_Vs==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_VEKTORENVERSCHUB,
MF_CHECKED);
                if(sw_Fxy==1)   o.CheckMenuItem(ID_ANSICHT_FUNKTIONSMATRIXFXY, MF_CHECKED);
                if(sw_Theta==1) o.CheckMenuItem(ID_ANSICHT_THETAFENSTERQ, MF_CHECKED);
                if(sw_Log==1)   o.CheckMenuItem(ID_ANSICHT_GRP2LOG, MF_CHECKED);
                if(sw_status==1)o.CheckMenuItem(ID_ANSICHT_STATUSLEISTE, MF_CHECKED);

                          SetMenu(&o);

                if(filename!=filename_tmp) { filename=filename_tmp; MINMAX();fn_THETA_1();} //fx
matrix


                sw_mod_=1;

                sw_FK=0; sw_xy=1; mod_FnP=1; linB_FnP=2;//keine kurve,  xy punkte rund

                sw_v0=3;

                sw_inv=1; OnModusInv();
        }
}

void CGRP2Dlg::OnModusFzx() //f(zx)
{
        if(sw_mod_!=5)
        {
                                CMenu o;
                                              o.LoadMenu(IDR_MENU1);
                if(sw_inv==1)
                {

        o.ModifyMenu(ID_MODUS_FX,MF_BYCOMMAND,ID_MODUS_FX,"f-1(x)=f(y)");

        o.ModifyMenu(ID_MODUS_FZX,MF_BYCOMMAND,ID_MODUS_FZX,"f-1(zx)=f(zy)");

        o.ModifyMenu(ID_MODUS_RXY,MF_BYCOMMAND,ID_MODUS_RXY,"r(y,x)");

        o.ModifyMenu(ID_MODUS_FP,MF_BYCOMMAND,ID_MODUS_FP,"F-1(p)=F(q)");

        o.ModifyMenu(ID_MODUS_EPSILON,MF_BYCOMMAND,ID_MODUS_EPSILON,"F-1(e)=F(p)");
                                              o.CheckMenuItem(ID_MODUS_INV,  MF_CHECKED);
                }
                                              o.CheckMenuItem(ID_MODUS_FX,  MF_UNCHECKED);
                                              o.CheckMenuItem(ID_MODUS_FP,  MF_UNCHECKED);
                         o.CheckMenuItem(ID_MODUS_FZX,  MF_CHECKED);  //
                                              o.CheckMenuItem(ID_MODUS_RXY, MF_UNCHECKED);
                                              o.CheckMenuItem(ID_MODUS_EPSILON,  MF_UNCHECKED);
                if(sw_inv==0)   o.CheckMenuItem(ID_MODUS_INV,  MF_UNCHECKED);
                if(sw_inv==1)   o.CheckMenuItem(ID_MODUS_INV,  MF_CHECKED);
                if(sw_csr==1)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_WEISS,  MF_CHECKED);
                if(sw_csr==2)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_SCHWARZ,  MF_CHECKED);
                if(sw_Fxy==1)   o.CheckMenuItem(ID_ANSICHT_FUNKTIONSMATRIXFXY, MF_CHECKED);
                if(sw_Theta==1) o.CheckMenuItem(ID_ANSICHT_THETAFENSTERQ, MF_CHECKED);
                if(sw_Log==1)   o.CheckMenuItem(ID_ANSICHT_GRP2LOG, MF_CHECKED);
                if(sw_status==1)o.CheckMenuItem(ID_ANSICHT_STATUSLEISTE, MF_CHECKED);

                          SetMenu(&o);

                if(filename!=filename_tmp) { filename=filename_tmp; MINMAX();fn_THETA_1();} //fx
matrix

                fn_THETA_2();//z-Matrix
```

```
               filename_tmp=filename;
                                       filename=filename_z;
               MINMAX();
               fn_THETA_1();//kennwerte

               sw_FK=0; sw_xy=1; mod_FnP=1; linB_FnP=2;//keine kurve,  xy punkte rund

               sw_v0=3;

               sw_mod_=5; sw_inv=1; OnModusInv();
       }
}


void CGRP2Dlg::OnModusRxy() //r(x,y)
{
       if(sw_mod_!=2)
       {
                                CMenu o;
                                             o.LoadMenu(IDR_MENU1);
               if(sw_inv==1)
               {

       o.ModifyMenu(ID_MODUS_FX,MF_BYCOMMAND,ID_MODUS_FX,"f-1(x)=f(y)");

       o.ModifyMenu(ID_MODUS_FZX,MF_BYCOMMAND,ID_MODUS_FZX,"f-1(zx)=f(zy)");

       o.ModifyMenu(ID_MODUS_RXY,MF_BYCOMMAND,ID_MODUS_RXY,"r(y,x)");

       o.ModifyMenu(ID_MODUS_FP,MF_BYCOMMAND,ID_MODUS_FP,"F-1(p)=F(q)");

       o.ModifyMenu(ID_MODUS_EPSILON,MF_BYCOMMAND,ID_MODUS_EPSILON,"F-1(e)=F(p)");
                                             o.CheckMenuItem(ID_MODUS_INV,  MF_CHECKED);
               }
                                             o.CheckMenuItem(ID_MODUS_FX,  MF_UNCHECKED);
                                             o.CheckMenuItem(ID_MODUS_FZX,  MF_UNCHECKED);
                                             o.CheckMenuItem(ID_MODUS_FP,  MF_UNCHECKED);
                                             o.CheckMenuItem(ID_MODUS_RXY, MF_CHECKED); //
                                             o.CheckMenuItem(ID_MODUS_EPSILON,  MF_UNCHECKED);
               if(sw_inv==0)   o.CheckMenuItem(ID_MODUS_INV,  MF_UNCHECKED);
               if(sw_inv==1)   o.CheckMenuItem(ID_MODUS_INV,  MF_CHECKED);
               if(sw_csr==1)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_WEISS,  MF_CHECKED);
               if(sw_csr==2)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_SCHWARZ,  MF_CHECKED);
               if(sw_Fxy==1)   o.CheckMenuItem(ID_ANSICHT_FUNKTIONSMATRIXFXY, MF_CHECKED);
               if(sw_Theta==1) o.CheckMenuItem(ID_ANSICHT_THETAFENSTERQ, MF_CHECKED);
               if(sw_Log==1)   o.CheckMenuItem(ID_ANSICHT_GRP2LOG, MF_CHECKED);
               if(sw_status==1)o.CheckMenuItem(ID_ANSICHT_STATUSLEISTE, MF_CHECKED);

                         SetMenu(&o);

               if(filename!=filename_tmp) { filename=filename_tmp; MINMAX();fn_THETA_1();} //fx
matrix

               fn_THETA_2();//z-Matrix

               filename_tmp=filename;
                                       filename=filename_z;
               MINMAX();
               //fn_THETA_1();//kennwerte

               sw_FK=0; sw_xy=1; mod_FnP=2; linB_FnP=1;//keine kurve,  xy punkte eckig

               sw_v0=1;

               sw_mod_=2; sw_inv=1; OnModusInv();
       }
}

//menue befehle
void CGRP2Dlg::OnModusFp() //F(p)
{
       if(sw_mod_!=3)
       {
                                CMenu o;
                                             o.LoadMenu(IDR_MENU1);
               if(sw_inv==1)
               {
```

```
        o.ModifyMenu(ID_MODUS_FX,MF_BYCOMMAND,ID_MODUS_FX,"f-1(x)=f(y)");

        o.ModifyMenu(ID_MODUS_FZX,MF_BYCOMMAND,ID_MODUS_FZX,"f-1(zx)=f(zy)");

        o.ModifyMenu(ID_MODUS_RXY,MF_BYCOMMAND,ID_MODUS_RXY,"r(y,x)");

        o.ModifyMenu(ID_MODUS_FP,MF_BYCOMMAND,ID_MODUS_FP,"F-1(p)=F(q)");

        o.ModifyMenu(ID_MODUS_EPSILON,MF_BYCOMMAND,ID_MODUS_EPSILON,"F-1(e)=F(p)");
                                              o.CheckMenuItem(ID_MODUS_INV,  MF_CHECKED);
                }
                                              o.CheckMenuItem(ID_MODUS_FX,  MF_UNCHECKED);
                                              o.CheckMenuItem(ID_MODUS_FZX,  MF_UNCHECKED);
                                              o.CheckMenuItem(ID_MODUS_FP,  MF_CHECKED);//
                                              o.CheckMenuItem(ID_MODUS_RXY, MF_UNCHECKED);
                                              o.CheckMenuItem(ID_MODUS_EPSILON,  MF_UNCHECKED);
                if(sw_inv==0)   o.CheckMenuItem(ID_MODUS_INV,  MF_UNCHECKED);
                if(sw_inv==1)   o.CheckMenuItem(ID_MODUS_INV,  MF_CHECKED);
                if(sw_csr==1)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_WEISS,  MF_CHECKED);
                if(sw_csr==2)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_SCHWARZ,  MF_CHECKED);
                if(sw_As==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_ACHSENVERSCHUB,
MF_CHECKED);
            if(sw_Vs==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_VEKTORENVERSCHUB,
MF_CHECKED);
                if(sw_Fxy==1)   o.CheckMenuItem(ID_ANSICHT_FUNKTIONSMATRIXFXY, MF_CHECKED);
                if(sw_Theta==1) o.CheckMenuItem(ID_ANSICHT_THETAFENSTERQ, MF_CHECKED);
                if(sw_Log==1)   o.CheckMenuItem(ID_ANSICHT_GRP2LOG, MF_CHECKED);
                if(sw_status==1)o.CheckMenuItem(ID_ANSICHT_STATUSLEISTE, MF_CHECKED);

                           SetMenu(&o);

                if(filename!=filename_tmp) { filename=filename_tmp; MINMAX();fn_THETA_1();} //fx
matrix

                fn_THETA_3(sw_pq);//p-Matrix

                filename_tmp=filename;
                                       filename=filename_p;

                MINMAX();
                fn_THETA_1();//kennwerte
                fn_THETA_4();


                sw_FK=1; sw_xy=0; // kurve,  keine xy punkte

                if(sw_mod_==2){sw_mod_=3;sw_inv=1; OnModusInv();} //bei rxy inv

                sw_v0=2;

                sw_mod_=3;

                sw_mkoord_A=1;mAx=min_x;mAy=min_y; // achsenposition
                sw_mkoord_V=1;mVx=0.5;mVy=(min_y+max_y)/2; // vektorposition
                if(sw_inv==1){mVy=0.5;mVx=(min_x+max_x)/2;}

                RedrawWindow();

                if(sw_mod_!=2)if(sw_Theta){sw_Theta=0;OnAnsichtThetafensterq(1);} //Theta
Ansicht aktualisieren
                if(sw_mod_!=2)if(sw_Fxy){sw_Fxy=0;OnAnsichtFunktionsmatrixfxy(1);} //Fxy
Ansicht aktualisieren
                          if(sw_Log){sw_Log=0;OnAnsichtGrp2log(1);} //Log Ansicht aktualisieren

        }
}


void CGRP2Dlg::OnModusEpsilon() // F(e)
{
        if(sw_mod_!=4)
        {
                                CMenu o;
                                              o.LoadMenu(IDR_MENU1);
                if(sw_inv==1)
                {
```

```
        o.ModifyMenu(ID_MODUS_FX,MF_BYCOMMAND,ID_MODUS_FX,"f-1(x)=f(y)");

        o.ModifyMenu(ID_MODUS_FZX,MF_BYCOMMAND,ID_MODUS_FZX,"f-1(zx)=f(zy)");

        o.ModifyMenu(ID_MODUS_RXY,MF_BYCOMMAND,ID_MODUS_RXY,"r(y,x)");

        o.ModifyMenu(ID_MODUS_FP,MF_BYCOMMAND,ID_MODUS_FP,"F-1(p)=F(q)");

        o.ModifyMenu(ID_MODUS_EPSILON,MF_BYCOMMAND,ID_MODUS_EPSILON,"F-1(e)=F(p)");
                                             o.CheckMenuItem(ID_MODUS_INV,  MF_CHECKED);
                }
                                             o.CheckMenuItem(ID_MODUS_FX,  MF_UNCHECKED);
                                             o.CheckMenuItem(ID_MODUS_FZX, MF_UNCHECKED);
                                             o.CheckMenuItem(ID_MODUS_FP,  MF_UNCHECKED);
                                             o.CheckMenuItem(ID_MODUS_RXY, MF_UNCHECKED);
                                             o.CheckMenuItem(ID_MODUS_EPSILON,  MF_CHECKED); //
                if(sw_inv==0)   o.CheckMenuItem(ID_MODUS_INV,  MF_UNCHECKED);
                if(sw_inv==1)   o.CheckMenuItem(ID_MODUS_INV,  MF_CHECKED);
                if(sw_csr==1)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_WEISS,  MF_CHECKED);
                if(sw_csr==2)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_SCHWARZ,  MF_CHECKED);
            if(sw_As==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_ACHSENVERSCHUB,
MF_CHECKED);
            if(sw_Vs==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_VEKTORENVERSCHUB,
MF_CHECKED);
                if(sw_Fxy==1)   o.CheckMenuItem(ID_ANSICHT_FUNKTIONSMATRIXFXY, MF_CHECKED);
                if(sw_Theta==1) o.CheckMenuItem(ID_ANSICHT_THETAFENSTERQ, MF_CHECKED);
                if(sw_Log==1)   o.CheckMenuItem(ID_ANSICHT_GRP2LOG, MF_CHECKED);
                if(sw_status==1)o.CheckMenuItem(ID_ANSICHT_STATUSLEISTE, MF_CHECKED);

                          SetMenu(&o);

                if(filename!=filename_tmp) { filename=filename_tmp; MINMAX();fn_THETA_1();} //fx
matrix


                fn_THETA_5();//p-Matrix epsilon

                filename_tmp=filename;
                                      filename=filename_p;

                MINMAX();
                //fn_THETA_1();//kennwerte

                sw_FK=1; sw_xy=0; //  kurve,   keine xy punkte zu bearbeiten...

                sw_mod_=4; sw_inv=1; OnModusInv();

                sw_v0=4; //vektor ursprung

                sw_mkoord_A=1;mAx=min_x;mAy=min_y; // achsenposition
                sw_mkoord_V=1;mVx=qE.sw;mVy=max_y; // vektorposition
                if(sw_inv==1){mVy=qE.sw;mVx=max_x;}

                RedrawWindow();
        }
}

void CGRP2Dlg::OnModusInv() //invers
{
                  CMenu o;
                        o.LoadMenu(IDR_MENU1);
        if(sw_inv==0)
        {
                                      o.ModifyMenu(ID_MODUS_FX,MF_BYCOMMAND,ID_MODUS_FX,"f-
1(x)=f(y)");
                                      o.ModifyMenu(ID_MODUS_FZX,MF_BYCOMMAND,ID_MODUS_FZX,"f-
1(zx)=f(zy)");

        o.ModifyMenu(ID_MODUS_RXY,MF_BYCOMMAND,ID_MODUS_RXY,"r(y,x)");
                                      o.ModifyMenu(ID_MODUS_FP,MF_BYCOMMAND,ID_MODUS_FP,"F-
1(p)=F(q)");

        o.ModifyMenu(ID_MODUS_EPSILON,MF_BYCOMMAND,ID_MODUS_EPSILON,"F-1(e)=F(p)");
                                      o.CheckMenuItem(ID_MODUS_INV,  MF_CHECKED);
        }
        if(sw_mod_==1)  o.CheckMenuItem(ID_MODUS_FX,  MF_CHECKED);
        if(sw_mod_==2)  o.CheckMenuItem(ID_MODUS_RXY,  MF_CHECKED);
        if(sw_mod_==3)  o.CheckMenuItem(ID_MODUS_FP,  MF_CHECKED);
        if(sw_mod_==4)  o.CheckMenuItem(ID_MODUS_EPSILON,  MF_CHECKED);
        if(sw_mod_==5)  o.CheckMenuItem(ID_MODUS_FZX,  MF_CHECKED);
        if(sw_csr==1)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_WEISS,  MF_CHECKED);
        if(sw_csr==2)   o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_SCHWARZ,  MF_CHECKED);
        if(sw_As==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_ACHSENVERSCHUB, MF_CHECKED);
        if(sw_Vs==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_VEKTORENVERSCHUB, MF_CHECKED);
        if(sw_Fxy==1)   o.CheckMenuItem(ID_ANSICHT_FUNKTIONSMATRIXFXY, MF_CHECKED);
        if(sw_Theta==1) o.CheckMenuItem(ID_ANSICHT_THETAFENSTERQ, MF_CHECKED);
        if(sw_Log==1)   o.CheckMenuItem(ID_ANSICHT_GRP2LOG, MF_CHECKED);
        if(sw_status==1)o.CheckMenuItem(ID_ANSICHT_STATUSLEISTE, MF_CHECKED);

                  SetMenu(&o);

        sw_inv=!sw_inv;
        swli=1;
        MINMAX();
                                    sw_mkoord_A=1;mAx=min_x; mAy=min_y;   // achsenposition
allgemein
                                                           sw_mkoord_V=1;mVx=max_x;mVy=max_y;
// vektorposition

        if(sw_mod_==2)             {sw_mkoord_A=1;mAy=0;   mAx=0;} // achsenposition rxy
        if(sw_mod_==2)             {sw_mkoord_V=1;mVy=1;   mVx=1;} // vektorposition rxy

        if(sw_mod_==3)             {sw_mkoord_V=1;mVy=0.5;   mVx=(min_x+max_x)/2;} //
vektorposition Fp
        if(sw_mod_==3)if(sw_inv==0){sw_mkoord_V=1;mVx=0.5;   mVy=(min_y+max_y)/2;} //
vektorposition Fp
        if(sw_mod_==4)             {sw_mkoord_V=1;mVy=qE.sw;   mVx=max_x;} // vektorposition Fe
        if(sw_mod_==4)if(sw_inv==0){sw_mkoord_V=1;mVx=qE.sw;   mVy=max_y;} // vektorposition Fe

        RedrawWindow();

        if(sw_Theta){sw_Theta=0;OnAnsichtThetafensterq(1);} //Theta Ansicht aktualisieren
        if(sw_Fxy){sw_Fxy=0;OnAnsichtFunktionsmatrixfxy(1);} //Fxy Ansicht aktualisieren
}

void CGRP2Dlg::OnEinstellungenCursorSchwarz()
{
                        CMenu o;
                                     o.LoadMenu(IDR_MENU1);
                                     o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_WEISS,
MF_UNCHECKED);
                                     o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_SCHWARZ,
MF_CHECKED);
        if(sw_inv==1)
        {
                                     o.ModifyMenu(ID_MODUS_FX,MF_BYCOMMAND,ID_MODUS_FX,"f-
1(x)=f(y)");
                                     o.ModifyMenu(ID_MODUS_FZX,MF_BYCOMMAND,ID_MODUS_FZX,"f-
1(zx)=f(zy)");

        o.ModifyMenu(ID_MODUS_RXY,MF_BYCOMMAND,ID_MODUS_RXY,"r(y,x)");
                                     o.ModifyMenu(ID_MODUS_FP,MF_BYCOMMAND,ID_MODUS_FP,"F-
1(p)=F(q)");

        o.ModifyMenu(ID_MODUS_EPSILON,MF_BYCOMMAND,ID_MODUS_EPSILON,"F-1(e)=F(p)");
                                     o.CheckMenuItem(ID_MODUS_INV,  MF_CHECKED);
        }
        if(sw_mod_==1)  o.CheckMenuItem(ID_MODUS_FX,  MF_CHECKED);
        if(sw_mod_==2)  o.CheckMenuItem(ID_MODUS_RXY,  MF_CHECKED);
        if(sw_mod_==3)  o.CheckMenuItem(ID_MODUS_FP,  MF_CHECKED);
        if(sw_mod_==4)  o.CheckMenuItem(ID_MODUS_EPSILON,  MF_CHECKED);
        if(sw_mod_==5)  o.CheckMenuItem(ID_MODUS_FZX,  MF_CHECKED);
        if(sw_As==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_ACHSENVERSCHUB, MF_CHECKED);
        if(sw_Vs==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_VEKTORENVERSCHUB, MF_CHECKED);
        if(sw_Fxy==1)   o.CheckMenuItem(ID_ANSICHT_FUNKTIONSMATRIXFXY, MF_CHECKED);
        if(sw_Theta==1) o.CheckMenuItem(ID_ANSICHT_THETAFENSTERQ, MF_CHECKED);
        if(sw_Log==1)   o.CheckMenuItem(ID_ANSICHT_GRP2LOG, MF_CHECKED);
        if(sw_status==1)o.CheckMenuItem(ID_ANSICHT_STATUSLEISTE, MF_CHECKED);


           SetMenu(&o);

   if(sw_emf_in==0)sw_csr=2; OnMouseMove(0,0) ;
}
```

```
void CGRP2Dlg::OnEinstellungenCursorWeiss()
{
                        CMenu o;
                                     o.LoadMenu(IDR_MENU1);
                                     o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_WEISS,
MF_CHECKED);
                                     o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_SCHWARZ,
MF_UNCHECKED);
        if(sw_inv==1)
        {
                                     o.ModifyMenu(ID_MODUS_FX,MF_BYCOMMAND,ID_MODUS_FX,"f-
1(x)=f(y)");
                                     o.ModifyMenu(ID_MODUS_FZX,MF_BYCOMMAND,ID_MODUS_FZX,"f-
1(zx)=f(zy)");

        o.ModifyMenu(ID_MODUS_RXY,MF_BYCOMMAND,ID_MODUS_RXY,"r(y,x)");
                                     o.ModifyMenu(ID_MODUS_FP,MF_BYCOMMAND,ID_MODUS_FP,"F-
1(p)=F(q)");

        o.ModifyMenu(ID_MODUS_EPSILON,MF_BYCOMMAND,ID_MODUS_EPSILON,"F-1(e)=F(p)");
                                     o.CheckMenuItem(ID_MODUS_INV,  MF_CHECKED);
        }
        if(sw_mod_==1)  o.CheckMenuItem(ID_MODUS_FX,  MF_CHECKED);
        if(sw_mod_==2)  o.CheckMenuItem(ID_MODUS_RXY,  MF_CHECKED);
        if(sw_mod_==3)  o.CheckMenuItem(ID_MODUS_FP,  MF_CHECKED);
        if(sw_mod_==4)  o.CheckMenuItem(ID_MODUS_EPSILON,  MF_CHECKED);
        if(sw_mod_==5)  o.CheckMenuItem(ID_MODUS_FZX,  MF_CHECKED);
        if(sw_As==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_ACHSENVERSCHUB, MF_CHECKED);
        if(sw_Vs==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_VEKTORENVERSCHUB, MF_CHECKED);
        if(sw_Fxy==1)   o.CheckMenuItem(ID_ANSICHT_FUNKTIONSMATRIXFXY, MF_CHECKED);
        if(sw_Theta==1) o.CheckMenuItem(ID_ANSICHT_THETAFENSTERQ, MF_CHECKED);
        if(sw_Log==1)   o.CheckMenuItem(ID_ANSICHT_GRP2LOG, MF_CHECKED);
        if(sw_status==1)o.CheckMenuItem(ID_ANSICHT_STATUSLEISTE, MF_CHECKED);


               SetMenu(&o);

   sw_csr=1; OnMouseMove(0,0) ;
}

void CGRP2Dlg::OnEinstellungenCursor_() //menue modifikation bei individual-cursor schalter
{
                        CMenu o;
                                     o.LoadMenu(IDR_MENU1);
                                     o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_WEISS,
MF_UNCHECKED);
        if(!csr_)             o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_SCHWARZ,  MF_UNCHECKED);
        if(csr_)              o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_SCHWARZ,  MF_CHECKED);
        if(sw_inv==1)
        {
                                     o.ModifyMenu(ID_MODUS_FX,MF_BYCOMMAND,ID_MODUS_FX,"f-
1(x)=f(y)");
                                     o.ModifyMenu(ID_MODUS_FZX,MF_BYCOMMAND,ID_MODUS_FX,"f-
1(zx)=f(zy)");

        o.ModifyMenu(ID_MODUS_RXY,MF_BYCOMMAND,ID_MODUS_RXY,"r(y,x)");
                                     o.ModifyMenu(ID_MODUS_FP,MF_BYCOMMAND,ID_MODUS_FP,"F-
1(p)=F(q)");

        o.ModifyMenu(ID_MODUS_EPSILON,MF_BYCOMMAND,ID_MODUS_EPSILON,"F-1(e)=F(p)");
                                     o.CheckMenuItem(ID_MODUS_INV,  MF_CHECKED);
        }
        if(sw_mod_==1)  o.CheckMenuItem(ID_MODUS_FX,  MF_CHECKED);
        if(sw_mod_==2)  o.CheckMenuItem(ID_MODUS_RXY,  MF_CHECKED);
        if(sw_mod_==3)  o.CheckMenuItem(ID_MODUS_FP,  MF_CHECKED);
        if(sw_mod_==4)  o.CheckMenuItem(ID_MODUS_EPSILON,  MF_CHECKED);
        if(sw_mod_==5)  o.CheckMenuItem(ID_MODUS_FZX,  MF_CHECKED);
        if(!csr_)             o.EnableMenuItem(ID_EINSTELLUNGEN_CURSOR_WEISS, MF_GRAYED );
        if(!csr_)             o.EnableMenuItem(ID_EINSTELLUNGEN_CURSOR_SCHWARZ, MF_GRAYED );
        if(sw_As==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_ACHSENVERSCHUB, MF_CHECKED);
        if(sw_Vs==1)    o.CheckMenuItem(ID_EINSTELLUNGEN_CURSOR_VEKTORENVERSCHUB, MF_CHECKED);
        if(sw_Fxy==1)   o.CheckMenuItem(ID_ANSICHT_FUNKTIONSMATRIXFXY, MF_CHECKED);
        if(sw_Theta==1) o.CheckMenuItem(ID_ANSICHT_THETAFENSTERQ, MF_CHECKED);
        if(sw_Log==1)   o.CheckMenuItem(ID_ANSICHT_GRP2LOG, MF_CHECKED);
        if(sw_status==1)o.CheckMenuItem(ID_ANSICHT_STATUSLEISTE, MF_CHECKED);

           SetMenu(&o);
   OnMouseMove(0,0) ;
}


void CGRP2Dlg::OnClose()
{
        GetWindowRect(&coord); //fensterrechteckkoordinaten in RECT strukturpointer coord

        CWinApp*   pApp = AfxGetApp(); // ini profil schreiben

                   pApp->WriteProfileInt("Achsen","x_Beschriftung_Werte",sw_x);
                      pApp->WriteProfileInt("Achsen","y_Beschriftung_Werte",sw_y);
                         pApp->WriteProfileInt("Achsen","x_Beschriftung_MinMax",sw_xm);
                   pApp->WriteProfileInt("Achsen","y_Beschriftung_MinMax",sw_ym);
                         pApp->WriteProfileInt("Achsen","x_",sw_xA);
                   pApp->WriteProfileInt("Achsen","y_",sw_yA);
                   pApp->WriteProfileInt("Achsen","x_Vektor",sw_xV);
                   pApp->WriteProfileInt("Achsen","y_Vektor",sw_yV);
                   pApp->WriteProfileInt("Achsen","x_Skala",sw_xS);
                   pApp->WriteProfileInt("Achsen","y_Skala",sw_yS);
                         pApp->WriteProfileInt("Achsen","x_Skalenwerte",sw_xSw);
                   pApp->WriteProfileInt("Achsen","y_Skalenwerte",sw_ySw);
                         pApp->WriteProfileInt("Achsen","x_Koordinate",sw_xK);
                   pApp->WriteProfileInt("Achsen","y_Koordinate",sw_yK);
               pApp->WriteProfileInt("Funktion","Kurve",sw_FK);
                   pApp->WriteProfileInt("Funktion","Punkte",sw_xy);
                         pApp->WriteProfileString("Funktion","Datei",filename_tmp);
                         pApp->WriteProfileString("EMF","Datei",emf_filename);
                         pApp-
>WriteProfileString("Dialog","Diagramm_Skalierung_1",ftoc(sc,8));
                   pApp->WriteProfileString("Dialog","Diagramm_Skalierung_2",ftoc(scy,8));
                         pApp->WriteProfileInt("Dialog","Emf_Skalierung_x",scrx);
                         pApp->WriteProfileInt("Dialog","Emf_Skalierung_y",scry);
        if(wnd_pos)pApp->WriteProfileInt("Dialog","Position_x",coord.left);
        if(wnd_pos)pApp->WriteProfileInt("Dialog","Position_y",coord.top);
        if(wnd_pos)pApp->WriteProfileInt("Dialog","Grösse_x",dlg.x);
        if(wnd_pos)pApp->WriteProfileInt("Dialog","Grösse_y",dlg.y);
                         pApp->WriteProfileString("Dialog","Vektor_x",ftoc(mv1x,8));
                   pApp->WriteProfileString("Dialog","Vektor_y",ftoc(mv1y,8));
                         pApp->WriteProfileString("Dialog","Achse_x",ftoc(mv2x,8));
                   pApp->WriteProfileString("Dialog","Achse_y",ftoc(mv2y,8));
                         pApp->WriteProfileString("Achsen","x_Justierung",ftoc(corx*100,8));
                   pApp->WriteProfileString("Achsen","y_Justierung",ftoc(cory*100,8));
                         pApp->WriteProfileString("Achsen","x_Skala_Teilung",ftoc(tlg_x,8));
                   pApp->WriteProfileString("Achsen","y_Skala_Teilung",ftoc(tlg_y,8));
                         pApp->WriteProfileInt("EMF","öffnen",sw_emf_in);
                         pApp->WriteProfileInt("Programm","Funktionsmatrixfenster",sw_Fxy);
                         pApp->WriteProfileInt("Programm","Thetafenster",sw_Theta);
                         pApp->WriteProfileInt("Programm","Logfenster",sw_Log);
                         pApp->WriteProfileInt("Programm","Statusleiste", sw_status);
                         pApp->WriteProfileInt("Programm","Cursor_Art",sw_csr);
                   pApp->WriteProfileInt("Programm","Achsen_Verschub",sw_As);
                   pApp->WriteProfileInt("Programm","Vektoren_Verschub",sw_Vs);
                 if(1)pApp->WriteProfileInt("Programm","Startmodus",sw_mod_);


        CDialog::OnClose();
}

//steuerfunktionen
void CGRP2Dlg::_FnKu(int x){sw_FK=x;}
void CGRP2Dlg::_xyPkt(int y){sw_xy=y;}
void CGRP2Dlg::sc_(CString x){sc =atof(x);}
void CGRP2Dlg::sc_y(CString y){scy=atof(y);}
void CGRP2Dlg::x_scala(int x){sw_x=x;}                void CGRP2Dlg::y_scala(int y){sw_y=y;}
void CGRP2Dlg::x_Sw(int x){sw_xSw=x;}                 void CGRP2Dlg::y_Sw(int y){sw_ySw=y;}
void CGRP2Dlg::x_minmax(int x){sw_xm=x;}              void CGRP2Dlg::y_minmax(int y){sw_ym=y;}
void CGRP2Dlg::x_Achse(int x){sw_xA=x;}               void CGRP2Dlg::y_Achse(int y){sw_yA=y;}
void CGRP2Dlg::x_Vektor(int x){sw_xV=x;}              void CGRP2Dlg::y_Vektor(int y){sw_yV=y;}
void CGRP2Dlg::x_Skala(int x){sw_xS=x;}               void CGRP2Dlg::y_Skala(int y){sw_yS=y;}
void CGRP2Dlg::x_Koor(int x){sw_xK=x;}                void CGRP2Dlg::y_Koor(int y){sw_yK=y;}
void CGRP2Dlg::x_kj(CString x){corx=atof(x)/100;}     void CGRP2Dlg::y_kj(CString
y){cory=atof(y)/100;}
void CGRP2Dlg::x_tlg(CString x){tlg_x=atof(x);}       void CGRP2Dlg::y_tlg(CString
y){tlg_y=atof(y);}
void CGRP2Dlg::x_emf(CString x){scrx =atoi(x);}       void CGRP2Dlg::y_emf(CString
y){scry=atoi(y);}
void CGRP2Dlg::xK_ds(int x){ds_xk=x;}                 void CGRP2Dlg::yK_ds(int y){ds_yk=y;}
void CGRP2Dlg::xS_ds(int x){ds_xSw=x;}               void CGRP2Dlg::yS_ds(int y){ds_ySw=y;}
void CGRP2Dlg::xW_ds(int x){ds_x=x;}                 void CGRP2Dlg::yW_ds(int y){ds_y=y;}
void CGRP2Dlg::clr_hg(int fb){fb_hg=fb;};
void CGRP2Dlg::clr_K(int fb){fb_K=fb;};
void CGRP2Dlg::clr_P(int fb){fb_P=fb;};
void CGRP2Dlg::clr_X(int fb){fb_Ax=fb;};
void CGRP2Dlg::clr_Y(int fb){fb_Ay=fb;};
void CGRP2Dlg::clr_Xgrd(int fb){fb_Grdx=fb;};       void CGRP2Dlg::clr_Ygrd(int
fb){fb_Grdy=fb;};
void CGRP2Dlg::clr_Xv(int fb){fb_Vx=fb;};           void CGRP2Dlg::clr_Yv(int fb){fb_Vy=fb;};
void CGRP2Dlg::K_gr(int gr){linB_Fn=gr;};
void CGRP2Dlg::P_gr(int gr){linB_FnP=gr;};
void CGRP2Dlg::X_gr(int gr){linB_Ax=gr;};           void CGRP2Dlg::Y_gr(int gr){linB_Ay=gr;};
void CGRP2Dlg::Xv_gr(int gr){linB_Vx=gr;};          void CGRP2Dlg::Yv_gr(int gr){linB_Vy=gr;};
void CGRP2Dlg::Xgrd_gr(int gr){linB_Grdx=gr;};      void CGRP2Dlg::Ygrd_gr(int
gr){linB_Grdy=gr;};
void CGRP2Dlg::fn_font_x(CString font,int h, int w, int fb){ fn_x_fon=font; fn_x_H=h;
fn_x_W=w; fn_x_fb=fb; }
void CGRP2Dlg::fn_font_y(CString font,int h, int w, int fb){ fn_y_fon=font; fn_y_H=h;
fn_y_W=w; fn_y_fb=fb; }
void CGRP2Dlg::font_Ax(  CString font,int h, int w, int fb){ Ax_fon=  font; Ax_H=  h; Ax_W=
w; Ax_fb=  fb; }
void CGRP2Dlg::font_Ay(  CString font,int h, int w, int fb){ Ay_fon=  font; Ay_H=  h; Ay_W=
w; Ay_fb=  fb; }
void CGRP2Dlg::font_V(   CString font,int h, int w, int fb){ V_fon=   font; V_H=   h; V_W=
w; V_fb=   fb; }
void CGRP2Dlg::P_art(CString art){    if(art=="rund") mod_FnP=1;  if(art=="eckig")mod_FnP=2;
if(art=="Kreuz")mod_FnP=3; };
void CGRP2Dlg::K_art(CString art){    if(art=="Linie")mod_Fn=1;   if(art=="Punkt")mod_Fn=2;
if(art=="Strich")mod_Fn=3;   if(art=="Strichpunkt")mod_Fn=4;
if(art=="Strichdoppelpkt.")mod_Fn=5;   };
void CGRP2Dlg::X_art(CString art){    if(art=="Linie")mod_Ax=1;   if(art=="Punkt")mod_Ax=2;
if(art=="Strich")mod_Ax=3;   if(art=="Strichpunkt")mod_Ax=4;
if(art=="Strichdoppelpkt.")mod_Ax=5;   };
void CGRP2Dlg::Y_art(CString art){    if(art=="Linie")mod_Ay=1;   if(art=="Punkt")mod_Ay=2;
if(art=="Strich")mod_Ay=3;   if(art=="Strichpunkt")mod_Ay=4;
if(art=="Strichdoppelpkt.")mod_Ay=5;   };
void CGRP2Dlg::Xv_art(CString art){   if(art=="Linie")mod_Vx=1;   if(art=="Punkt")mod_Vx=2;
if(art=="Strich")mod_Vx=3;   if(art=="Strichpunkt")mod_Vx=4;
if(art=="Strichdoppelpkt.")mod_Vx=5;   };
void CGRP2Dlg::Yv_art(CString art){   if(art=="Linie")mod_Vy=1;   if(art=="Punkt")mod_Vy=2;
if(art=="Strich")mod_Vy=3;   if(art=="Strichpunkt")mod_Vy=4;
if(art=="Strichdoppelpkt.")mod_Vy=5;   };
void CGRP2Dlg::X_gridart(CString art){if(art=="Linie")mod_Grdx=1; if(art=="Punkt")mod_Grdx=2;
if(art=="Strich")mod_Grdx=3; if(art=="Strichpunkt")mod_Grdx=4;
if(art=="Strichdoppelpkt.")mod_Grdx=5; };
void CGRP2Dlg::Y_gridart(CString art){if(art=="Linie")mod_Grdy=1; if(art=="Punkt")mod_Grdy=2;
if(art=="Strich")mod_Grdy=3; if(art=="Strichpunkt")mod_Grdy=4;
if(art=="Strichdoppelpkt.")mod_Grdy=5; };
void CGRP2Dlg::rxy_art(CString art){  if(art=="Linie")mod_r=1;    if(art=="Punkt")mod_r=2;
if(art=="Strich")mod_r=3;    if(art=="Strichpunkt")mod_r=4;
if(art=="Strichdoppelpkt.")mod_r=5;    };
void CGRP2Dlg::ryx_art(CString art){  if(art=="Linie")mod_ri=1;   if(art=="Punkt")mod_ri=2;
if(art=="Strich")mod_ri=3;   if(art=="Strichpunkt")mod_ri=4;
if(art=="Strichdoppelpkt.")mod_ri=5;   };
void CGRP2Dlg::s1xy_art(CString art){ if(art=="Linie")mod_Grdx=1; if(art=="Punkt")mod_Grdx=2;
if(art=="Strich")mod_Grdx=3; if(art=="Strichpunkt")mod_Grdx=4;
if(art=="Strichdoppelpkt.")mod_Grdx=5; };
void CGRP2Dlg::s1yx_art(CString art){ if(art=="Linie")mod_sgyx=1; if(art=="Punkt")mod_sgyx=2;
if(art=="Strich")mod_sgyx=3; if(art=="Strichpunkt")mod_sgyx=4;
if(art=="Strichdoppelpkt.")mod_sgyx=5; };
void CGRP2Dlg::srx_art(CString art){  if(art=="Linie")mod_sR=1;   if(art=="Punkt")mod_sR=2;
if(art=="Strich")mod_sR=3;   if(art=="Strichpunkt")mod_sR=4;
if(art=="Strichdoppelpkt.")mod_sR=5;   };
void CGRP2Dlg::srxy_art(CString art){ if(art=="Linie")mod_sr=1;     if(art=="Punkt")mod_sr=2;
if(art=="Strich")mod_sr=3;   if(art=="Strichpunkt")mod_sr=4;
if(art=="Strichdoppelpkt.")mod_sr=5;   };
void CGRP2Dlg::sry_art(CString art){  if(art=="Linie")mod_sRi=1;  if(art=="Punkt")mod_sRi=2;
if(art=="Strich")mod_sRi=3;  if(art=="Strichpunkt")mod_sRi=4;
if(art=="Strichdoppelpkt.")mod_sRi=5;  };
void CGRP2Dlg::sryx_art(CString art){ if(art=="Linie")mod_sri=1;  if(art=="Punkt")mod_sri=2;
if(art=="Strich")mod_sri=3;  if(art=="Strichpunkt")mod_sri=4;
if(art=="Strichdoppelpkt.")mod_sri=5;  };
void CGRP2Dlg::sxy_art(CString art){  if(art=="Linie")mod_sxy=1;  if(art=="Punkt")mod_sxy=2;
if(art=="Strich")mod_sxy=3;  if(art=="Strichpunkt")mod_sxy=4;
if(art=="Strichdoppelpkt.")mod_sxy=5;  };
```

```
void CGRP2Dlg::syx_art(CString art){  if(art=="Linie")mod_syx=1;  if(art=="Punkt")mod_syx=2;
if(art=="Strich")mod_syx=3;  if(art=="Strichpunkt")mod_syx=4;
if(art=="Strichdoppelpkt.")mod_syx=5;  };
void CGRP2Dlg::a3_art(CString art){   if(art=="Linie")mod_a3=1;   if(art=="Punkt")mod_a3=2;
         if(art=="Strich")mod_a3=3;   if(art=="Strichpunkt")mod_a3=4;
if(art=="Strichdoppelpkt.")mod_a3=5;   };
void CGRP2Dlg::a4_art(CString art){    if(art=="Linie")mod_a4=1;     if(art=="Punkt")mod_a4=2;
         if(art=="Strich")mod_a4=3;   if(art=="Strichpunkt")mod_a4=4;
if(art=="Strichdoppelpkt.")mod_a4=5;   };
void CGRP2Dlg::ag3_art(CString art){  if(art=="Linie")mod_ag3=1;  if(art=="Punkt")mod_ag3=2;
if(art=="Strich")mod_ag3=3;  if(art=="Strichpunkt")mod_ag3=4;
if(art=="Strichdoppelpkt.")mod_ag3=5;  };
void CGRP2Dlg::ag4_art(CString art){  if(art=="Linie")mod_ag4=1;  if(art=="Punkt")mod_ag4=2;
if(art=="Strich")mod_ag4=3;  if(art=="Strichpunkt")mod_ag4=4;
if(art=="Strichdoppelpkt.")mod_ag4=5;  };
void CGRP2Dlg::sa3g_art(CString art){ if(art=="Linie")mod_sga3=1; if(art=="Punkt")mod_sga3=2;
if(art=="Strich")mod_sga3=3; if(art=="Strichpunkt")mod_sga3=4;
if(art=="Strichdoppelpkt.")mod_sga3=5; };
void CGRP2Dlg::sa4g_art(CString art){ if(art=="Linie")mod_sga4=1; if(art=="Punkt")mod_sga4=2;
if(art=="Strich")mod_sga4=3; if(art=="Strichpunkt")mod_sga4=4;
if(art=="Strichdoppelpkt.")mod_sga4=5; };
void CGRP2Dlg::sd_art(CString art){    if(art=="Linie")mod_sd=1;     if(art=="Punkt")mod_sd=2;
         if(art=="Strich")mod_sd=3;   if(art=="Strichpunkt")mod_sd=4;
if(art=="Strichdoppelpkt.")mod_sd=5;   };
void CGRP2Dlg::sdg_art(CString art){  if(art=="Linie")mod_sdg=1;  if(art=="Punkt")mod_sdg=2;
if(art=="Strich")mod_sdg=3;  if(art=="Strichpunkt")mod_sdg=4;
if(art=="Strichdoppelpkt.")mod_sdg=5;  };
void CGRP2Dlg::sgx_art(CString art){  if(art=="Linie")mod_sgam=1; if(art=="Punkt")mod_sgam=2;
if(art=="Strich")mod_sgam=3; if(art=="Strichpunkt")mod_sgam=4;
if(art=="Strichdoppelpkt.")mod_sgam=5; };
void CGRP2Dlg::Xp_art(CString art){    if(art=="Linie")mod_am=1;     if(art=="Punkt")mod_am=2;
         if(art=="Strich")mod_am=3;   if(art=="Strichpunkt")mod_am=4;
if(art=="Strichdoppelpkt.")mod_am=5;   };
void CGRP2Dlg::e_art(CString art){    if(art=="Linie")mod_e=1;    if(art=="Punkt")mod_e=2;
if(art=="Strich")mod_e=3;       if(art=="Strichpunkt")mod_e=4;
if(art=="Strichdoppelpkt.")mod_e=5;    };
void CGRP2Dlg::s0_art(CString art){   if(art=="Linie")mod_s0=1;     if(art=="Punkt")mod_s0=2;
         if(art=="Strich")mod_s0=3;   if(art=="Strichpunkt")mod_s0=4;
if(art=="Strichdoppelpkt.")mod_s0=5;   };
void CGRP2Dlg::s1_art(CString art){   if(art=="Linie")mod_s1=1;   if(art=="Punkt")mod_s1=2;
if(art=="Strich")mod_s1=3;   if(art=="Strichpunkt")mod_s1=4;
if(art=="Strichdoppelpkt.")mod_s1=5;   };
void CGRP2Dlg::xe_art(CString art){   if(art=="Linie")mod_x=1;      if(art=="Punkt")mod_x=2;
         if(art=="Strich")mod_x=3;      if(art=="Strichpunkt")mod_x=4;
if(art=="Strichdoppelpkt.")mod_x=5;    };
void CGRP2Dlg::x0_art(CString art){    if(art=="Linie")mod_x0=1;     if(art=="Punkt")mod_x0=2;
         if(art=="Strich")mod_x0=3;   if(art=="Strichpunkt")mod_x0=4;
if(art=="Strichdoppelpkt.")mod_x0=5;   };
void CGRP2Dlg::x1_art(CString art){    if(art=="Linie")mod_x1=1;     if(art=="Punkt")mod_x1=2;
         if(art=="Strich")mod_x1=3;   if(art=="Strichpunkt")mod_x1=4;
if(art=="Strichdoppelpkt.")mod_x1=5;   };
void CGRP2Dlg::D_rxy(int sw){rxy_D=sw;};
void CGRP2Dlg::r_rxy(int sw){rxy_ =sw;};            void CGRP2Dlg::r_ryx(int sw){ryx_ =sw;};
void CGRP2Dlg::r_sxy(int sw){sxy_ =sw;};            void CGRP2Dlg::r_syx(int sw){syx_ =sw;};
void CGRP2Dlg::r_s1xy(int sw){s1xy_ =sw;};          void CGRP2Dlg::r_s1yx(int sw){s1yx_ =sw;};
void CGRP2Dlg::r_srxy(int sw){srxy_ =sw;};          void CGRP2Dlg::r_sryx(int sw){sryx_ =sw;};
void CGRP2Dlg::r_srx(int sw){srx_ =sw;};            void CGRP2Dlg::r_sry(int sw){sry_ =sw;};
void CGRP2Dlg::p_q_sw(short sw){ sw_pq =sw;};//sw_v0=2;sw_mkoord_V=1;mVy=0.5;
mVx=(min_x+max_x)/2;
void CGRP2Dlg::x_Grid(short sw){sw_Grdx =sw;};      void CGRP2Dlg::y_Grid(short sw){sw_Grdy
=sw;};
void CGRP2Dlg::p_a3(short sw){a3_ =sw;};            void CGRP2Dlg::p_a4(short sw ){a4_ =sw;};
void CGRP2Dlg::p_ag3(short sw ){ag3_ =sw;};         void CGRP2Dlg::p_ag4(short sw ){ag4_
=sw;};
void CGRP2Dlg::p_sga3(short sw ){sga3_ =sw;};       void CGRP2Dlg::p_sga4(short sw ){sga4_
=sw;};
void CGRP2Dlg::p_am(short sw ){am_ =sw;};
void CGRP2Dlg::p_sd(short sw ){sd_ =sw;};
void CGRP2Dlg::p_sdg(short sw ){sdg_ =sw;};
void CGRP2Dlg::p_sgam(short sw ){sgam_=sw;};
void CGRP2Dlg::CI_Fp(double p)// Konfidenzintervall Fp
{
                                      CI_Pp=p;
       ci_zp=zp_funktion((100 -((100-CI_Pp)/2))/100);
       ci_tp=tp_funktion((100 -((100-CI_Pp)/2))/100,itof(n_));
};
void CGRP2Dlg::CI_rxy(double p)// Konfidenzintervall rxy
{
```

```
                                      CI_Pr=p;
       ci_zr=zp_funktion((100 -((100-CI_Pr)/2))/100);
       ci_tr=tp_funktion((100 -((100-CI_Pr)/2))/100,itof(n_-2));
};
void CGRP2Dlg::CI_Fe(BOOL sw, double p, int df)// Konfidenzintervall Fe
{
                                             CI_Pe=p;
       if(!sw)ci_ze=zp_funktion((100 -((100-CI_Pe)/2))/100);
       if( sw)ci_ze=tp_funktion((100 -((100-CI_Pe)/2))/100,itof(df));
};

void CGRP2Dlg::rnd_dyn(short sw){dynrnd  =sw;};
void CGRP2Dlg::rnd_fsr(short sw){filestr =sw;};
void CGRP2Dlg::w_pos_(short sw){wnd_pos  =sw;};
void CGRP2Dlg::csr_sw(short sw){csr_ =sw;};
void CGRP2Dlg::m_AX_(float x){mAx =x;};             void CGRP2Dlg::m_AY_(float y){mAy =y;};
void CGRP2Dlg::m_VX_(float x){mVx =x;};             void CGRP2Dlg::m_VY_(float y){mVy =y;};
void CGRP2Dlg::posVx_(float x){posVx =x;};          void CGRP2Dlg::posVy_(float y){posVy =y;};
void CGRP2Dlg::posAXx_(float x){posAXx =x;};        void CGRP2Dlg::posAXy_(float y){posAXy
=y;};
void CGRP2Dlg::posAYx_(float x){posAYx =x;};        void CGRP2Dlg::posAYy_(float y){posAYy
=y;};
void CGRP2Dlg::posFXx_(float x){posFXx =x;};        void CGRP2Dlg::posFXy_(float y){posFXy
=y;};
void CGRP2Dlg::posFYx_(float x){posFYx =x;};        void CGRP2Dlg::posFYy_(float y){posFYy
=y;};
void CGRP2Dlg::clr_rxy(int fb){fb_r=fb;};           void CGRP2Dlg::clr_ryx(int fb){fb_ri=fb;};
void CGRP2Dlg::clr_s1xy(int fb){fb_sgxy=fb;};       void CGRP2Dlg::clr_s1yx(int
fb){fb_sgyx=fb;};
void CGRP2Dlg::clr_srx(int fb){fb_sR=fb;};          void CGRP2Dlg::clr_sry(int
fb){fb_sRi=fb;};
void CGRP2Dlg::clr_srxy(int fb){fb_sr=fb;}          void CGRP2Dlg::clr_sryx(int
fb){fb_sri=fb;};;
void CGRP2Dlg::clr_syx(int fb){fb_syx=fb;};         void CGRP2Dlg::clr_sxy(int
fb){fb_sxy=fb;};
void CGRP2Dlg::clr_a3(int fb){fb_a3=fb;};           void CGRP2Dlg::clr_a4(int fb){fb_a4=fb;};
void CGRP2Dlg::clr_ag3(int fb){fb_ag3=fb;};         void CGRP2Dlg::clr_ag4(int
fb){fb_ag4=fb;};
void CGRP2Dlg::clr_sa3g(int fb){fb_sga3=fb;};       void CGRP2Dlg::clr_sa4g(int
fb){fb_sga4=fb;};
void CGRP2Dlg::clr_sd(int fb){fb_sd=fb;};
void CGRP2Dlg::clr_sdg(int fb){fb_sdg=fb;};
void CGRP2Dlg::clr_sgx(int fb){fb_sgam=fb;};
void CGRP2Dlg::clr_Xp(int fb){fb_am=fb;};
void CGRP2Dlg::rxy_gr(int gr){linB_r=gr;};          void CGRP2Dlg::ryx_gr(int
gr){linB_ri=gr;};
void CGRP2Dlg::s1xy_gr(int gr){linB_sgxy=gr;};      void CGRP2Dlg::s1yx_gr(int
gr){linB_sgyx=gr;};
void CGRP2Dlg::srx_gr(int gr){linB_sR=gr;};         void CGRP2Dlg::sry_gr(int
gr){linB_sRi=gr;};
void CGRP2Dlg::srxy_gr(int gr){linB_sr=gr;};        void CGRP2Dlg::sryx_gr(int
gr){linB_sri=gr;};
void CGRP2Dlg::syx_gr(int gr){linB_syx=gr;};        void CGRP2Dlg::sxy_gr(int
gr){linB_sxy=gr;};
void CGRP2Dlg::a3_gr(int gr){linB_a3=gr;};          void CGRP2Dlg::a4_gr(int gr){linB_a4=gr;};
void CGRP2Dlg::ag3_gr(int gr){linB_ag3=gr;};        void CGRP2Dlg::ag4_gr(int
gr){linB_ag4=gr;};
void CGRP2Dlg::sa3g_gr(int gr){linB_sga3=gr;};      void CGRP2Dlg::sa4g_gr(int
gr){linB_sga4=gr;};
void CGRP2Dlg::sd_gr(int gr){linB_sd=gr;};
void CGRP2Dlg::sdg_gr(int gr){linB_sdg=gr;};
void CGRP2Dlg::sgx_gr(int gr){linB_sgam=gr;};
void CGRP2Dlg::Xp_gr(int gr){linB_am=gr;};
void CGRP2Dlg::q_rxy(short sw){r_q=sw;};            void CGRP2Dlg::q_ryx(short sw){ri_q=sw;};
void CGRP2Dlg::q_sgyx(short sw){sgyx_q=sw;};        void CGRP2Dlg::q_sgxy(short
sw){sgxy_q=sw;};
void CGRP2Dlg::q_srx(short sw){sR_q=sw;};           void CGRP2Dlg::q_sry(short sw){sRi_q=sw;};
void CGRP2Dlg::q_srxy(short sw){sr_q=sw;};          void CGRP2Dlg::q_sryx(short
sw){sri_q=sw;};
void CGRP2Dlg::q_syx(short sw){syx_q=sw;};          void CGRP2Dlg::q_sxy(short sw){sxy_q=sw;};
void CGRP2Dlg::q_a3(short sw){a3_q=sw;};            void CGRP2Dlg::q_a4(short sw){a4_q=sw;};
void CGRP2Dlg::q_ag3(short sw){ag3_q=sw;};          void CGRP2Dlg::q_ag4(short sw){ag4_q=sw;};
void CGRP2Dlg::q_sa3g(short sw){sa3g_q=sw;};        void CGRP2Dlg::q_sa4g(short
sw){sa4g_q=sw;};
void CGRP2Dlg::q_sd(short sw){sd_q=sw;};
void CGRP2Dlg::q_sdg(short sw){sdg_q=sw;};
void CGRP2Dlg::q_sgam(short sw){sgam_q=sw;};
void CGRP2Dlg::q_am(short sw){am_q=sw;};
void CGRP2Dlg::X_bez(short sw){sw_xb =sw;};         void CGRP2Dlg::Y_bez(short sw){sw_yb
=sw;};
void CGRP2Dlg::sw_p_e(short sw){p_e_sw =sw;};
void CGRP2Dlg::sw_p_p(short sw){p_p_sw =sw;};
void CGRP2Dlg::pos_scx_(short sw){posSCx =sw;};     void CGRP2Dlg::pos_scy_(short sw){posSCy
=sw;};
void CGRP2Dlg::pos_Axx_(short sw){posBXx =sw;};     void CGRP2Dlg::pos_Axy_(short sw){posBXy
=sw;};
void CGRP2Dlg::pos_Ayx_(short sw){posBYx =sw;};     void CGRP2Dlg::pos_Ayy_(short sw){posBYy
=sw;};
void CGRP2Dlg::pos_Dx_(short sw){posX =sw;};        void CGRP2Dlg::pos_Dy_(short sw){posY
=sw;};
void CGRP2Dlg::frm_Dx_(float sw){frmX =sw;};        void CGRP2Dlg::frm_Dy_(float sw){frmY
=sw;};


void CGRP2Dlg::e_gr(int gr) { linB_e=gr ;};
void CGRP2Dlg::s0_gr(int gr) {linB_s0 =gr ;};       void CGRP2Dlg::s1_gr(int gr) { linB_s1=
gr;};
void CGRP2Dlg::xe_gr(int gr) { linB_x= gr;};
void CGRP2Dlg::x0_gr(int gr) { linB_x0=gr ;};       void CGRP2Dlg::x1_gr(int gr) { linB_x1=
gr;};
void CGRP2Dlg::clr_e(int fb)  { fb_e=fb ;};
void CGRP2Dlg::clr_s0(int fb)  { fb_s0=fb ;};       void CGRP2Dlg::clr_s1(int fb)  { fb_s1=fb
;};
void CGRP2Dlg::clr_xe(int fb) { fb_x=fb ;};
void CGRP2Dlg::clr_x0(int fb) { fb_x0=fb ;};        void CGRP2Dlg::clr_x1(int fb) { fb_x1=fb
;};
void CGRP2Dlg::q_e(short sw){ e_q=sw ;};
void CGRP2Dlg::q_s0(short sw){ s0_q=sw;};           void CGRP2Dlg::q_s1(short sw){ s1_q=sw ;};
void CGRP2Dlg::q_xe(short sw){x_q=sw ;};
void CGRP2Dlg::q_x0(short sw){ x0_q=sw ;};          void CGRP2Dlg::q_x1(short sw){ x1_q=sw ;};
void CGRP2Dlg::p_e(short sw ) { e_= sw; };
void CGRP2Dlg::p_s0(short sw ){ s0_= sw;};          void CGRP2Dlg::p_s1(short sw ){ s1_= sw;};
void CGRP2Dlg::p_x0(short sw ){ x0_= sw;};          void CGRP2Dlg::p_x1(short sw ){ x1_= sw;};
void CGRP2Dlg::p_x(short sw)  { xc_= sw;};
void CGRP2Dlg::AxBez_(short sw, CString x){ sw_xbz=sw; xBz=x; }; void CGRP2Dlg::AyBez_(short
sw, CString y)  { sw_ybz=sw; yBz=y; };

void CGRP2Dlg::int_sw( int n){sw_integral=1;     if(n==0)sw_integral=2;     int_n=n; }
//integralschaltfunktion
void CGRP2Dlg::diff_sw(int n){sw_differential=1; if(n==0)sw_differential=2; diff_n=n;}
//differentialschaltfunktion
void CGRP2Dlg::kgl_sw(int i) {sw_kgl=1;          if(i==0)sw_kgl=2;          kgl_i=i; }
//mittelungsschaltfunktion
void CGRP2Dlg::fxy_sw( ) {    sw_fxy=1;                                               }
//funktionsmatrix neu setzenschaltfunktion

//////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////
//
//
//
//
//
//                             //       ////      ////
//                             //      //  //    //
//                             //      //  //    //  //
//                             ////     ///       /////
//
//
//
//
//
//
//////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////
void CGRP2Dlg::log_file(BOOL sw) //Logfile
{
            char datum[9];
       _strdate( datum );

            char zeit[9];
       _strtime( zeit );

                   FILE *f;
                         f = fopen (log_filename, "a");
```

```
                 fprintf(f,"%s  %s  %i:\t",datum,zeit,nlog);

       if(sw==1){fprintf(f,"[%s]->(F)\n",filename_tmp);} //funktonsmatrix in
       if(sw==2){fprintf(f,"[%s]<-(F)\n",filename_tmp);} //funktonsmatrix out








              fclose( f );

       nlog++;
}


//EnhancedMetafile EMF
//////////////////////////////////////////////////////////////////////////////////////////////
///////////////////////////////////////////////////
//
//
//
//
//
//                             //////     ///  ///     //////
//                             //         // // //     //
//                             ////        //    //     ////
//                             //          //    //     //
//                             //////     //    //     //
//
//
//
//
//
//
//////////////////////////////////////////////////////////////////////////////////////////////
///////////////////////////////////////////////////
void CGRP2Dlg::GRP_Diagramm()
{
       CMetaFileDC ooo;
                                          CRect og1(0, 0,dlg.x*scrx, dlg.y*scry);
                    ooo.CreateEnhanced(0,emfname,og1,"GRP-Diagramm");

       SetCursor(m_Csr3); //

       //////////////////////////////////////////////////////////////////////////////////////
///////

               if(fb_hg!=13357270) //hintergrundfarbe nicht windowsgrau
               {
                      //UpdateWindow();
                                            CRect rect(0, 0,dlg.x, dlg.y);
                      ooo.FillSolidRect(rect,fb_hg);
               }

               // schriftartendefinition
               CFont of1;of1.CreateFont(Ay_H,  Ay_W,
0,0,400,0,0,0,0,OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,DEFAULT_PITCH,Ay_fon
);//y Achse
               CFont of2;of2.CreateFont(Ax_H,  Ax_W,
0,0,400,0,0,0,0,OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,DEFAULT_PITCH,Ax_fon
);//x Achse
           CFont
ofx;ofx.CreateFont(fn_x_H,fn_x_W,0,0,400,0,0,0,0,OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAUL
T_QUALITY,DEFAULT_PITCH,fn_x_fon);//x funktionswerte
               CFont
ofy;ofy.CreateFont(fn_y_H,fn_y_W,0,0,400,0,0,0,0,OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAUL
T_QUALITY,DEFAULT_PITCH,fn_y_fon);//y funktionswerte
               CFont ofv;ofv.CreateFont(V_H,   V_W,
0,0,400,0,0,0,0,OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,DEFAULT_PITCH,V_fon
);//xy Vektor

               //linienartdefinition
                              CPen oos;
               if(mod_Ay==1)oos.CreatePen(PS_SOLID,      linB_Ay,fb_Ay); //y achse
               if(mod_Ay==2)oos.CreatePen(PS_DOT,        linB_Ay,fb_Ay); //...
               if(mod_Ay==3)oos.CreatePen(PS_DASH,      linB_Ay,fb_Ay); //...
               if(mod_Ay==4)oos.CreatePen(PS_DASHDOT,   linB_Ay,fb_Ay); //...
               if(mod_Ay==5)oos.CreatePen(PS_DASHDOTDOT,linB_Ay,fb_Ay); //...
                       CPen o1s;
               if(mod_Ax==1)o1s.CreatePen(PS_SOLID,     linB_Ax,fb_Ax); //x achse
               if(mod_Ax==2)o1s.CreatePen(PS_DOT,       linB_Ax,fb_Ax); //...
               if(mod_Ax==3)o1s.CreatePen(PS_DASH,      linB_Ax,fb_Ax); //...
               if(mod_Ax==4)o1s.CreatePen(PS_DASHDOT,   linB_Ax,fb_Ax); //...
               if(mod_Ax==5)o1s.CreatePen(PS_DASHDOTDOT,linB_Ax,fb_Ax); //...
                       CPen o3s;
                            o3s.CreatePen(PS_SOLID,     linB_Ax,fb_Ax); //x achsen
teilungsstriche
                       CPen o4s;
                                   o4s.CreatePen(PS_SOLID,     linB_Ay,fb_Ay); //y achsen
teilungsstriche
                       CPen o2s;
               if(mod_Fn==1)o2s.CreatePen(PS_SOLID,     linB_Fn,fb_K); //funktionskurve
               if(mod_Fn==2)o2s.CreatePen(PS_DOT,       linB_Fn,fb_K);
               if(mod_Fn==3)o2s.CreatePen(PS_DASH,      linB_Fn,fb_K);
               if(mod_Fn==4)o2s.CreatePen(PS_DASHDOT,   linB_Fn,fb_K);
               if(mod_Fn==5)o2s.CreatePen(PS_DASHDOTDOT,linB_Fn,fb_K);
                       CPen o2s1;
                               o2s1.CreatePen(PS_SOLID,    linB_Fn,fb_P); //funktionspunkte
                       CPen o1;
        if(mod_Vx==1)o1.CreatePen(PS_SOLID,      linB_Vx,fb_Vx); //x vektor
               if(mod_Vx==2)o1.CreatePen(PS_DOT,        linB_Vx,fb_Vx); //...
               if(mod_Vx==3)o1.CreatePen(PS_DASH,       linB_Vx,fb_Vx); //...
               if(mod_Vx==4)o1.CreatePen(PS_DASHDOT,    linB_Vx,fb_Vx); //...
               if(mod_Vx==5)o1.CreatePen(PS_DASHDOTDOT, linB_Vx,fb_Vx); //...
                       CPen o2;
        if(mod_Vy==1)o2.CreatePen(PS_SOLID,      linB_Vy,fb_Vy); //y vektor
               if(mod_Vy==2)o2.CreatePen(PS_DOT,        linB_Vy,fb_Vy); //...
               if(mod_Vy==3)o2.CreatePen(PS_DASH,       linB_Vy,fb_Vy); //...
               if(mod_Vy==4)o2.CreatePen(PS_DASHDOT,    linB_Vy,fb_Vy); //...
               if(mod_Vy==5)o2.CreatePen(PS_DASHDOTDOT, linB_Vy,fb_Vy); //...
                       CPen or1;
               if(mod_r==1)or1.CreatePen(PS_SOLID,     linB_r,fb_r); //r
               if(mod_r==2)or1.CreatePen(PS_DOT,       linB_r,fb_r); //...
               if(mod_r==3)or1.CreatePen(PS_DASH,      linB_r,fb_r); //...
               if(mod_r==4)or1.CreatePen(PS_DASHDOT,   linB_r,fb_r); //...
               if(mod_r==5)or1.CreatePen(PS_DASHDOTDOT,linB_r,fb_r); //...
                          CPen or1i;
               if(mod_ri==1)or1i.CreatePen(PS_SOLID,     linB_ri,fb_ri); //ri
               if(mod_ri==2)or1i.CreatePen(PS_DOT,       linB_ri,fb_ri); //...
               if(mod_ri==3)or1i.CreatePen(PS_DASH,      linB_ri,fb_ri); //...
               if(mod_ri==4)or1i.CreatePen(PS_DASHDOT,   linB_ri,fb_ri); //...
               if(mod_ri==5)or1i.CreatePen(PS_DASHDOTDOT,linB_ri,fb_ri); //...
                             CPen or2;
               if(mod_sr==1)or2.CreatePen(PS_SOLID,     linB_sr,fb_sr); //sr
               if(mod_sr==2)or2.CreatePen(PS_DOT,       linB_sr,fb_sr); //...
               if(mod_sr==3)or2.CreatePen(PS_DASH,      linB_sr,fb_sr); //...
               if(mod_sr==4)or2.CreatePen(PS_DASHDOT,   linB_sr,fb_sr); //...
               if(mod_sr==5)or2.CreatePen(PS_DASHDOTDOT,linB_sr,fb_sr); //...
                          CPen or2i;
               if(mod_sri==1)or2i.CreatePen(PS_SOLID,     linB_sri,fb_sri); //sri
               if(mod_sri==2)or2i.CreatePen(PS_DOT,       linB_sri,fb_sri); //...
               if(mod_sri==3)or2i.CreatePen(PS_DASH,      linB_sri,fb_sri); //...
               if(mod_sri==4)or2i.CreatePen(PS_DASHDOT,   linB_sri,fb_sri); //...
               if(mod_sri==5)or2i.CreatePen(PS_DASHDOTDOT,linB_sri,fb_sri); //...
                             CPen or3;
               if(mod_sR==1)or3.CreatePen(PS_SOLID,     linB_sR,fb_sR); //sR
               if(mod_sR==2)or3.CreatePen(PS_DOT,       linB_sR,fb_sR); //...
               if(mod_sR==3)or3.CreatePen(PS_DASH,      linB_sR,fb_sR); //...
               if(mod_sR==4)or3.CreatePen(PS_DASHDOT,   linB_sR,fb_sR); //...
               if(mod_sR==5)or3.CreatePen(PS_DASHDOTDOT,linB_sR,fb_sR); //...
                          CPen or3i;
               if(mod_sRi==1)or3i.CreatePen(PS_SOLID,     linB_sRi,fb_sRi); //sRi
               if(mod_sRi==2)or3i.CreatePen(PS_DOT,       linB_sRi,fb_sRi); //...
               if(mod_sRi==3)or3i.CreatePen(PS_DASH,      linB_sRi,fb_sRi); //...
               if(mod_sRi==4)or3i.CreatePen(PS_DASHDOT,   linB_sRi,fb_sRi); //...
               if(mod_sRi==5)or3i.CreatePen(PS_DASHDOTDOT,linB_sRi,fb_sRi); //...
                             CPen or4i;
               if(mod_sxy==1)or4i.CreatePen(PS_SOLID,     linB_sxy,fb_sxy); //sxy
               if(mod_sxy==2)or4i.CreatePen(PS_DOT,       linB_sxy,fb_sxy); //...
               if(mod_sxy==3)or4i.CreatePen(PS_DASH,      linB_sxy,fb_sxy); //...
               if(mod_sxy==4)or4i.CreatePen(PS_DASHDOT,   linB_sxy,fb_sxy); //...
               if(mod_sxy==5)or4i.CreatePen(PS_DASHDOTDOT,linB_sxy,fb_sxy); //...
                          CPen or4;
```

```
if(mod_syx==1)or4.CreatePen(PS_SOLID,     linB_syx,fb_syx); //syx
if(mod_syx==2)or4.CreatePen(PS_DOT,       linB_syx,fb_syx); //...
if(mod_syx==3)or4.CreatePen(PS_DASH,      linB_syx,fb_syx); //...
if(mod_syx==4)or4.CreatePen(PS_DASHDOT,   linB_syx,fb_syx); //...
if(mod_syx==5)or4.CreatePen(PS_DASHDOTDOT,linB_syx,fb_syx); //...
             CPen or5i;
if(mod_sgxy==1)or5i.CreatePen(PS_SOLID,     linB_sgxy,fb_sgxy); //s'xy
if(mod_sgxy==2)or5i.CreatePen(PS_DOT,       linB_sgxy,fb_sgxy); //...
if(mod_sgxy==3)or5i.CreatePen(PS_DASH,      linB_sgxy,fb_sgxy); //...
if(mod_sgxy==4)or5i.CreatePen(PS_DASHDOT,   linB_sgxy,fb_sgxy); //...
if(mod_sgxy==5)or5i.CreatePen(PS_DASHDOTDOT,linB_sgxy,fb_sgxy); //...
               CPen or5;
if(mod_sgyx==1)or5.CreatePen(PS_SOLID,     linB_sgyx,fb_sgyx); //s'yx
if(mod_sgyx==2)or5.CreatePen(PS_DOT,       linB_sgyx,fb_sgyx); //...
if(mod_sgyx==3)or5.CreatePen(PS_DASH,      linB_sgyx,fb_sgyx); //...
if(mod_sgyx==4)or5.CreatePen(PS_DASHDOT,   linB_sgyx,fb_sgyx); //...
if(mod_sgyx==5)or5.CreatePen(PS_DASHDOTDOT,linB_sgyx,fb_sgyx); //...
        CPen op1;
if(mod_am==1)op1.CreatePen(PS_SOLID,     linB_am,fb_am); //am
if(mod_am==2)op1.CreatePen(PS_DOT,       linB_am,fb_am); //...
if(mod_am==3)op1.CreatePen(PS_DASH,      linB_am,fb_am); //...
if(mod_am==4)op1.CreatePen(PS_DASHDOT,   linB_am,fb_am); //...
if(mod_am==5)op1.CreatePen(PS_DASHDOTDOT,linB_am,fb_am); //...
               CPen op2;
if(mod_sd==1)op2.CreatePen(PS_SOLID,     linB_sd,fb_sd); //sd
if(mod_sd==2)op2.CreatePen(PS_DOT,       linB_sd,fb_sd); //...
if(mod_sd==3)op2.CreatePen(PS_DASH,      linB_sd,fb_sd); //...
if(mod_sd==4)op2.CreatePen(PS_DASHDOT,   linB_sd,fb_sd); //...
if(mod_sd==5)op2.CreatePen(PS_DASHDOTDOT,linB_sd,fb_sd); //...
           CPen op3;
if(mod_sgam==1)op3.CreatePen(PS_SOLID,     linB_sgam,fb_sgam); //s'am
if(mod_sgam==2)op3.CreatePen(PS_DOT,       linB_sgam,fb_sgam); //...
if(mod_sgam==3)op3.CreatePen(PS_DASH,      linB_sgam,fb_sgam); //...
if(mod_sgam==4)op3.CreatePen(PS_DASHDOT,   linB_sgam,fb_sgam); //...
if(mod_sgam==5)op3.CreatePen(PS_DASHDOTDOT,linB_sgam,fb_sgam); //...
        CPen op4;
if(mod_sdg==1)op4.CreatePen(PS_SOLID,     linB_sdg,fb_sdg); //sd'
if(mod_sdg==2)op4.CreatePen(PS_DOT,       linB_sdg,fb_sdg); //...
if(mod_sdg==3)op4.CreatePen(PS_DASH,      linB_sdg,fb_sdg); //...
if(mod_sdg==4)op4.CreatePen(PS_DASHDOT,   linB_sdg,fb_sdg); //...
if(mod_sdg==5)op4.CreatePen(PS_DASHDOTDOT,linB_sdg,fb_sdg); //...
        CPen op5;
if(mod_a3==1)op5.CreatePen(PS_SOLID,     linB_a3,fb_a3); //a3
if(mod_a3==2)op5.CreatePen(PS_DOT,       linB_a3,fb_a3); //...
if(mod_a3==3)op5.CreatePen(PS_DASH,      linB_a3,fb_a3); //...
if(mod_a3==4)op5.CreatePen(PS_DASHDOT,   linB_a3,fb_a3); //...
if(mod_a3==5)op5.CreatePen(PS_DASHDOTDOT,linB_a3,fb_a3); //...
        CPen op6;
if(mod_ag3==1)op6.CreatePen(PS_SOLID,     linB_ag3,fb_ag3); //a3'
if(mod_ag3==2)op6.CreatePen(PS_DOT,       linB_ag3,fb_ag3); //...
if(mod_ag3==3)op6.CreatePen(PS_DASH,      linB_ag3,fb_ag3); //...
if(mod_ag3==4)op6.CreatePen(PS_DASHDOT,   linB_ag3,fb_ag3); //...
if(mod_ag3==5)op6.CreatePen(PS_DASHDOTDOT,linB_ag3,fb_ag3); //...
        CPen op7;
if(mod_sga3==1)op7.CreatePen(PS_SOLID,     linB_sga3,fb_sga3); //s'a3
if(mod_sga3==2)op7.CreatePen(PS_DOT,       linB_sga3,fb_sga3); //...
if(mod_sga3==3)op7.CreatePen(PS_DASH,      linB_sga3,fb_sga3); //...
if(mod_sga3==4)op7.CreatePen(PS_DASHDOT,   linB_sga3,fb_sga3); //...
if(mod_sga3==5)op7.CreatePen(PS_DASHDOTDOT,linB_sga3,fb_sga3); //...
        CPen op8;
if(mod_a4==1)op8.CreatePen(PS_SOLID,     linB_a4,fb_a4); //a4
if(mod_a4==2)op8.CreatePen(PS_DOT,       linB_a4,fb_a4); //...
if(mod_a4==3)op8.CreatePen(PS_DASH,      linB_a4,fb_a4); //...
if(mod_a4==4)op8.CreatePen(PS_DASHDOT,   linB_a4,fb_a4); //...
if(mod_a4==5)op8.CreatePen(PS_DASHDOTDOT,linB_a4,fb_a4); //...
        CPen op9;
if(mod_ag4==1)op9.CreatePen(PS_SOLID,     linB_ag4,fb_ag4); //a4'
if(mod_ag4==2)op9.CreatePen(PS_DOT,       linB_ag4,fb_ag4); //...
if(mod_ag4==3)op9.CreatePen(PS_DASH,      linB_ag4,fb_ag4); //...
if(mod_ag4==4)op9.CreatePen(PS_DASHDOT,   linB_ag4,fb_ag4); //...
if(mod_ag4==5)op9.CreatePen(PS_DASHDOTDOT,linB_ag4,fb_ag4); //...
        CPen op10;
if(mod_sga4==1)op10.CreatePen(PS_SOLID,     linB_sga4,fb_sga4); //s'a4
if(mod_sga4==2)op10.CreatePen(PS_DOT,       linB_sga4,fb_sga4); //...
if(mod_sga4==3)op10.CreatePen(PS_DASH,      linB_sga4,fb_sga4); //...
if(mod_sga4==4)op10.CreatePen(PS_DASHDOT,   linB_sga4,fb_sga4); //...
if(mod_sga4==5)op10.CreatePen(PS_DASHDOTDOT,linB_sga4,fb_sga4); //...
       CPen op11;
```

```
               if(mod_e==1)op11.CreatePen(PS_SOLID,     linB_e,fb_e); //e
               if(mod_e==2)op11.CreatePen(PS_DOT,       linB_e,fb_e); //...
               if(mod_e==3)op11.CreatePen(PS_DASH,      linB_e,fb_e); //...
               if(mod_e==4)op11.CreatePen(PS_DASHDOT,   linB_e,fb_e); //...
               if(mod_e==5)op11.CreatePen(PS_DASHDOTDOT,linB_e,fb_e); //...
                      CPen op12;
               if(mod_x==1)op12.CreatePen(PS_SOLID,     linB_x,fb_x); //x
               if(mod_x==2)op12.CreatePen(PS_DOT,       linB_x,fb_x); //...
               if(mod_x==3)op12.CreatePen(PS_DASH,      linB_x,fb_x); //...
               if(mod_x==4)op12.CreatePen(PS_DASHDOT,   linB_x,fb_x); //...
               if(mod_x==5)op12.CreatePen(PS_DASHDOTDOT,linB_x,fb_x); //...
                       CPen op13;
               if(mod_x0==1)op13.CreatePen(PS_SOLID,     linB_x0,fb_x0); //x0
               if(mod_x0==2)op13.CreatePen(PS_DOT,       linB_x0,fb_x0); //...
               if(mod_x0==3)op13.CreatePen(PS_DASH,      linB_x0,fb_x0); //...
               if(mod_x0==4)op13.CreatePen(PS_DASHDOT,   linB_x0,fb_x0); //...
               if(mod_x0==5)op13.CreatePen(PS_DASHDOTDOT,linB_x0,fb_x0); //...
                       CPen op14;
               if(mod_x1==1)op14.CreatePen(PS_SOLID,     linB_x1,fb_x1); //x1
               if(mod_x1==2)op14.CreatePen(PS_DOT,       linB_x1,fb_x1); //...
               if(mod_x1==3)op14.CreatePen(PS_DASH,      linB_x1,fb_x1); //...
               if(mod_x1==4)op14.CreatePen(PS_DASHDOT,   linB_x1,fb_x1); //...
               if(mod_x1==5)op14.CreatePen(PS_DASHDOTDOT,linB_x1,fb_x1); //...
                       CPen op15;
               if(mod_s0==1)op15.CreatePen(PS_SOLID,     linB_s0,fb_s0); //s0
               if(mod_s0==2)op15.CreatePen(PS_DOT,       linB_s0,fb_s0); //...
               if(mod_s0==3)op15.CreatePen(PS_DASH,      linB_s0,fb_s0); //...
               if(mod_s0==4)op15.CreatePen(PS_DASHDOT,   linB_s0,fb_s0); //...
               if(mod_s0==5)op15.CreatePen(PS_DASHDOTDOT,linB_s0,fb_s0); //...
                       CPen op16;
               if(mod_s1==1)op16.CreatePen(PS_SOLID,     linB_s1,fb_s1); //s1
               if(mod_s1==2)op16.CreatePen(PS_DOT,       linB_s1,fb_s1); //...
               if(mod_s1==3)op16.CreatePen(PS_DASH,      linB_s1,fb_s1); //...
               if(mod_s1==4)op16.CreatePen(PS_DASHDOT,   linB_s1,fb_s1); //...
               if(mod_s1==5)op16.CreatePen(PS_DASHDOTDOT,linB_s1,fb_s1); //...
                                CPen ogx;
               if(mod_Grdx==1)ogx.CreatePen(PS_SOLID,     linB_Grdx,fb_Grdx); //gridx
               if(mod_Grdx==2)ogx.CreatePen(PS_DOT,       linB_Grdx,fb_Grdx); //...
               if(mod_Grdx==3)ogx.CreatePen(PS_DASH,      linB_Grdx,fb_Grdx); //...
               if(mod_Grdx==4)ogx.CreatePen(PS_DASHDOT,   linB_Grdx,fb_Grdx); //...
               if(mod_Grdx==5)ogx.CreatePen(PS_DASHDOTDOT,linB_Grdx,fb_Grdx); //...
                                CPen ogy;
               if(mod_Grdy==1)ogy.CreatePen(PS_SOLID,     linB_Grdy,fb_Grdy); //gridy
               if(mod_Grdy==2)ogy.CreatePen(PS_DOT,       linB_Grdy,fb_Grdy); //...
               if(mod_Grdy==3)ogy.CreatePen(PS_DASH,      linB_Grdy,fb_Grdy); //...
               if(mod_Grdy==4)ogy.CreatePen(PS_DASHDOT,   linB_Grdy,fb_Grdy); //...
               if(mod_Grdy==5)ogy.CreatePen(PS_DASHDOTDOT,linB_Grdy,fb_Grdy); //...


               //füllwerkzeugdefinition
               CBrush b1;b1.CreateSolidBrush(fb_hg);//xy punkt füllfarbe hintergrund
               CBrush b2;b2.CreateSolidBrush(fb_P );//xy punkt rechteck rahmenfarbe

               //rendering

               ooo.SelectObject(&oos);
               ooo.SelectObject(&of1);
               ooo.SetBkColor( fb_hg);//hintergrundfarbe
               ooo.SetTextColor(Ay_fb);
            //ooo.SetBkMode(TRANSPARENT);

                       float dx_s =  dlg.x/sc;                 // skalierte dialoggrösse x

                       float dx_sy = dlg.x/scy;                // skalierte dialoggrösse x

                       float dy_s =  dlg.y/sc;                 // skalierte dialoggrösse y

                       float dy_sy = dlg.y/scy;                // skalierte dialoggrösse y

                       //float e_x=1; // einheit x=1

                       float e_x=((min_x+sc0x)/(max_x+sc0x)); // einheit x

//float e_y=1; // einheit y=1

                       float e_y=((min_y+sc0y)/(max_y+sc0y)); // einheit y

                       float egx= e_x * dx_s ;                 // gewichtete einheit x
                      float egy= e_y * dy_s ;                // gewichtete einheit y


                                 float daptx = egx *(mv2x/(e_x/sc)) ; //dialog-achsen-
koordinatenpunkt x
                if(sw_mkoord_A) daptx = (((((mAx))  +sc0x)/(max_x+sc0x))* dx_s)+dx_sy;
//dialog-Achsen-koordinatenpunkt x manuell
                                              //~~~
                                    Ax_m= (((( daptx)-dx_sy)/dx_s)*(max_x+sc0x))-sc0x;;
//global für koordinatenübergabe an koordinateneinstellungsdialog
                if(!sw_mkoord_A)mAx=Ax_m;

                       if(sw_yA==1)
                    ooo.MoveTo( daptx*frmX+ posX, 0);         //y achse
                       if(sw_yA==1)
                       ooo.LineTo( daptx*frmX+ posX, dlg.y);     //

                                                                                     CString
ct_="y";
                                        if(sw_inv==1)
ct_="x";     //f-1(x)
                                                      if(sw_mod_==2)
ct_="z(y)";  //rxy
                                                      if(sw_mod_==3)if(sw_pq==0)
ct_="x=q";   //Fp
                                                      if(sw_mod_==3)if(sw_pq==1)
ct_="y=q";   //

if(sw_mod_==3)if(sw_inv==1)if(p_p_sw==0)if(sw_pq==0)ct_="p(x)";  //

if(sw_mod_==3)if(sw_inv==1)if(p_p_sw==0)if(sw_pq==1)ct_="p(y)";  //

if(sw_mod_==3)if(sw_inv==1)if(p_p_sw==1)if(sw_pq==0)ct_="pa1(x)";//

if(sw_mod_==3)if(sw_inv==1)if(p_p_sw==1)if(sw_pq==1)ct_="pa1(y)";//

if(sw_mod_==3)if(sw_inv==1)if(p_p_sw==2)if(sw_pq==0)ct_="pa2(x)";//

if(sw_mod_==3)if(sw_inv==1)if(p_p_sw==2)if(sw_pq==1)ct_="pa2(y)";//
                                                      if(sw_mod_==4)if(sw_inv==0)if(p_e_sw==0)
ct_="p";     //Fe
                                                      if(sw_mod_==4)if(sw_inv==0)if(p_e_sw==1)
ct_="pa1";   //
                                                      if(sw_mod_==4)if(sw_inv==0)if(p_e_sw==2)
ct_="pa2";   //
                                                      if(sw_mod_==4)if(sw_inv==1)
ct_="q";     //

                                                      if(sw_ybz)
ct_=yBz;    //manuell

                       if(sw_yA==1)
                       if(sw_yb>=1)ooo.TextOut(daptx*frmX-2+posBYx+ posX,        0+posBYy ,
ct_); //y achsen bezeichnung oben
                       if(sw_yA==1)
                       if(sw_yb==1)ooo.TextOut(daptx*frmX-2+posBYx+ posX, dlg.y-12+posBYy ,
ct_); //y achsen bezeichnung unten


                       ooo.SelectObject(&o1s);
                       ooo.SelectObject(&of2);
                       ooo.SetBkColor( fb_hg);//hintergrundfarbe
                    ooo.SetTextColor(Ax_fb);


                                    float dapty = egy  *(mv2y/(e_y/sc)) ; //dialog-achsen-
koordinatenpunkt y
                                 if(sw_mkoord_A)dapty = ( dlg.y -(((((mAy))) +sc0y)/(max_y+sc0y))*
dy_s)-dy_sy; //dialog-Achsen-koordinatenpunkt y manuell
                                                          //~~~
                                    if(!sw_mkoord_A)Ay_m= (((( egy*((1-mv2y)/(e_y/sc)))-
dy_sy)/dy_s)*(max_y+sc0y))-sc0y; //global für koordinatenübergabe an
koordinateneinstellungsdialog
                 //if( sw_mkoord_A)Ay_m=mAy;
                                    if(!sw_mkoord_A)     mAy=Ay_m;

                       if(sw_xA==1)
```

```
                        ooo.MoveTo( 0,     dapty*frmY+ posY) ;   //x achse
                        if(sw_xA==1)
                        ooo.LineTo( dlg.x, dapty*frmY+ posY);    //


                                                                            ct_="x";
                                                                                           if(sw_inv==1)
ct_="y";     //f-1(x)
                                                                                           if(sw_mod_==2)
ct_="z(x)";  //rxy

if(sw_mod_==3)if(p_p_sw==0)if(sw_pq==0) ct_="p(x)";  //Fp

if(sw_mod_==3)if(p_p_sw==0)if(sw_pq==1) ct_="p(y)";  //

if(sw_mod_==3)if(p_p_sw==1)if(sw_pq==0) ct_="pa1(x)";//

if(sw_mod_==3)if(p_p_sw==1)if(sw_pq==1) ct_="pa1(y)";//

if(sw_mod_==3)if(p_p_sw==2)if(sw_pq==0) ct_="pa2(x)";//

if(sw_mod_==3)if(p_p_sw==2)if(sw_pq==1) ct_="pa2(y)";//

if(sw_mod_==3)if(sw_inv==1)if(sw_pq==0) ct_="x=q";   //

if(sw_mod_==3)if(sw_inv==1)if(sw_pq==1) ct_="y=q";   //

if(sw_mod_==4)if(sw_inv==1)if(p_e_sw==0)ct_="p";     //Fe

if(sw_mod_==4)if(sw_inv==1)if(p_e_sw==1)ct_="pa1";   //

if(sw_mod_==4)if(sw_inv==1)if(p_e_sw==2)ct_="pa2";   //

if(sw_mod_==4)if(sw_inv==0)             ct_="q";     //

                                                                                           if(sw_xbz)
ct_=xBz;     //manuell
                        if(sw_xA==1)
                        if(sw_xb==1)ooo.TextOut(      0+posBXx , dapty*frmY-6+posBXy+ posY,
ct_); //x achsen bezeichnung links
                        if(sw_xA==1)
                        if(sw_xb>=1)ooo.TextOut(dlg.x-6+posBXx , dapty*frmY-6+posBXy+ posY ,
ct_); //x achsen bezeichnung rechts


        ////////////////////////////////////////////////////////////////////////////////
/////////////////

                if(sw_drw)//dynamisches rendern (rerender schalter)
                {
                        int i1=0;

                        int x; int y;

                        int yr;  int ys0; int ysi0; int yss0; int yssi0; int ysr0; int ysri0;
                        int yri; int ys1; int ysi1; int yss1; int yssi1; int ysr1; int ysri1;

                        float x_m_min; float y_m_min;
                        float x_m_max; float y_m_max;

                        float yr_m_min; float ys0_m_min;float ysi0_m_min;float yss0_m_min;float
yssi0_m_min;float ysr0_m_min;float ysri0_m_min;
                        float yr_m_max; float ys0_m_max;float ysi0_m_max;float yss0_m_max;float
yssi0_m_max;float ysr0_m_max;float ysri0_m_max;
                        float yri_m_min;float ys1_m_min;float ysi1_m_min;float yss1_m_min;float
yssi1_m_min;float ysr1_m_min;float ysri1_m_min;
                        float yri_m_max;float ys1_m_max;float ysi1_m_max;float yss1_m_max;float
yssi1_m_max;float ysr1_m_max;float ysri1_m_max;



        ////////////////////////////////////////////////////////////////////////////////
//////////////////////////
                        // Funktionswertschleife
/////////////////////////////////////////////////////////////////////////////////

        ////////////////////////////////////////////////////////////////////////////////
//////////////////////////
```

```
                      FILE *f_;
                                f_=fopen(filename,"r"); //über die funktionsmatrixdatei

                      if(filestr)do //filestream rendern
                      {
                                                                            char cx_[20],
cy_[20];

                             if(sw_inv==0||(sw_inv==1&&sw_mod_==2)) fscanf(f_,"%s%s",&cx_,
&cy_); //funktionswerte einlesen
                             if(sw_inv==1)          if(sw_mod_!=2)  fscanf(f_,"%s%s",&cy_,
&cx_); //f-1


                             x_=atof(cx_); x_+=sc0x;
                             y_=atof(cy_); y_+=sc0y;


                                            char ccx_[20];
                                      sprintf(ccx_,"%s",      ftoc(x_-sc0x, ds_x));
//formatierte skalenwerte x
                             if(0)sprintf(ccx_,"%g", atof(ftoc(x_-sc0x, ds_x)));//zu
implementieren...

                                            char ccy_[20];
                                      sprintf(ccy_,"%s",      ftoc(y_-sc0y, ds_y));
//formatierte skalenwerte y
                             if(0)sprintf(ccx_,"%g", atof(ftoc(x_-sc0x, ds_x)));//zu
implementieren...


                             //koordinatenpunkt berechnung

                                            float pr_x = x_/(max_x+sc0x); //xi proportion
                                            float pr_y = y_/(max_y+sc0y); //yi proportion

                              x=           pr_x * dx_s  ; //dialog koordinatenpunkt zu wert x
                              y= dlg.y - ( pr_y * dy_s ); //dialog koordinatenpunkt zu wert y

                              x+=dx_sy;
                              y-=dy_sy;


/////////////////////////////////////////////////////////////////////////////////////////////
/////////////
                              // THETA rxy berechnung
                              //

                              if(sw_mod_==2) //Regressionsgeradenwert y
                              {
                                                                      float pr_yr;
                                                                      float pr_yri;
                                       if(sw_inv==0||rxy_D==1)pr_yr  = ((atof(cx_)* qR )
+sc0y)/(max_y+sc0y); //xir   proportion
                                       if(sw_inv==1||rxy_D==1)pr_yri = ((atof(cy_)* qR )
+sc0x)/(max_x+sc0x); //xir-1 proportion

                                      if(sw_inv==0||rxy_D==1)yr= dlg.y - ( pr_yr * dy_s );
//dialog koordinatenpunkt y'x
                                      if(sw_inv==0||rxy_D==1)yr-=dy_sy;

                                      if(sw_inv==1||rxy_D==1)yri= pr_yri * dx_s;
//dialog koordinatenpunkt x'y
                                      if(sw_inv==1||rxy_D==1)yri+=dx_sy;
                              }

                              if(sw_mod_==2)if(1) //Standardvorhersagefehler sy'x
                              {
                                                                          float pr_ys0;
                                                                          float pr_ys1;
                                                                          float pr_ysi0;
                                                                          float pr_ysi1;
                                       if(sw_inv==0||rxy_D==1) pr_ys0 =  (( (atof(cx_)* qR )-
qS*ci_zr) +sc0y)/(max_y+sc0y); //sy'- proportion
                                       if(sw_inv==0||rxy_D==1) pr_ys1 =  (( (atof(cx_)* qR
)+qS*ci_zr) +sc0y)/(max_y+sc0y); //sy'+ proportion
```

```
                                       if(sw_inv==1||rxy_D==1) pr_ysi0 = (( (atof(cy_)* qR )-
qS*ci_zr) +sc0x)/(max_x+sc0x); //sx'- proportion
                                       if(sw_inv==1||rxy_D==1) pr_ysi1 = (( (atof(cy_)* qR
)+qS*ci_zr) +sc0x)/(max_x+sc0x); //sx'+ proportion

                                      if(sw_inv==0||rxy_D==1)
                                      {
                                             ys0= dlg.y - ( pr_ys0 * dy_s ); //dialog
koordinatenpunkt sy'x-
                                             ys0-=dy_sy;

                                             ys1= dlg.y - ( pr_ys1 * dy_s ); //dialog
koordinatenpunkt sy'x+
                                             ys1-=dy_sy;
                                      }

                                      if(sw_inv==1||rxy_D==1)
                                      {
                                             ysi0=  pr_ysi0 * dx_s ;           //dialog
koordinatenpunkt sx'y-
                                             ysi0+=dx_sy;

                                             ysi1=  pr_ysi1 * dx_s ;           //dialog
koordinatenpunkt sx'y+
                                             ysi1+=dx_sy;
                                      }
                               }

                               if(sw_mod_==2)if(1) //Geschätzter Standardvorhersagefehler
s'y'x
                               {
                                                                          float pr_yss0;
                                                                          float pr_yss1;
                                                                          float pr_yssi0;
                                                                          float pr_yssi1;
                                       if(sw_inv==0||rxy_D==1) pr_yss0  = (( (atof(cx_)* qR )-
qsS*ci_zr) +sc0y)/(max_y+sc0y); //sy'- proportion
                                       if(sw_inv==0||rxy_D==1) pr_yss1  = (( (atof(cx_)* qR
)+qsS*ci_zr) +sc0y)/(max_y+sc0y); //sy'+ proportion
                                       if(sw_inv==1||rxy_D==1) pr_yssi0 = (( (atof(cy_)* qR )-
qsS*ci_zr) +sc0x)/(max_x+sc0x); //sx'- proportion
                                       if(sw_inv==1||rxy_D==1) pr_yssi1 = (( (atof(cy_)* qR
)+qsS*ci_zr) +sc0x)/(max_x+sc0x); //sx'+ proportion

                                      if(sw_inv==0||rxy_D==1)
                                      {
                                             yss0= dlg.y - ( pr_yss0 * dy_s ); //dialog
koordinatenpunkt s'y'x-
                                             yss0-=dy_sy;

                                             yss1= dlg.y - ( pr_yss1 * dy_s ); //dialog
koordinatenpunkt s'y'x+
                                             yss1-=dy_sy;
                                      }

                                      if(sw_inv==1||rxy_D==1)
                                      {
                                             yssi0=  pr_yssi0 * dx_s ;           //dialog
koordinatenpunkt s'x'y-
                                             yssi0+= dx_sy;

                                             yssi1=  pr_yssi1 * dx_s ;           //dialog
koordinatenpunkt s'x'y+
                                             yssi1+= dx_sy;
                                      }
                               }

                               if(sw_mod_==2)if(1) //Geschätzter Standardfehler der Regression
s'r
                               {
                                                                      float pr_ysr0;
                                                                      float pr_ysr1;
                                                                      float pr_ysri0;
                                                                      float pr_ysri1;
                                      if(sw_inv==0||rxy_D==1) pr_ysr0 = ((atof(cx_)* (qR-
qsR*ci_tr )) +sc0y)/(max_y+sc0y); //xir'   proportion
                                      if(sw_inv==0||rxy_D==1) pr_ysr1 = ((atof(cx_)*
(qR+qsR*ci_tr )) +sc0y)/(max_y+sc0y); //xir'   proportion
```

```
                                      if(sw_inv==1||rxy_D==1) pr_ysri0 = ((atof(cy_)* (qR-
qsR*ci_tr )) +sc0x)/(max_x+sc0x); //xir'-1 proportion
                                      if(sw_inv==1||rxy_D==1) pr_ysri1 = ((atof(cy_)*
(qR+qsR*ci_tr )) +sc0x)/(max_x+sc0x); //xir'-1 proportion

                                      if(sw_inv==0||rxy_D==1)
                                      {
                                              ysr0= dlg.y - ( pr_ysr0 * dy_s ); //dialog
koordinatenpunkt y'x r'
                                              ysr0-=dy_sy;

                                              ysr1= dlg.y - ( pr_ysr1 * dy_s );
                                              ysr1-=dy_sy;
                                      }

                                      if(sw_inv==1||rxy_D==1)
                                      {
                                              ysri0= pr_ysri0 * dx_s;             //dialog
koordinatenpunkt x'y r'
                                              ysri0+=dx_sy;

                                              ysri1= pr_ysri1 * dx_s;
                                              ysri1+=dx_sy;
                                      }
                               }


/////////////////////////////////////////////////////////////////////////////////////////////
/////////////
                               // Funktionszeichnung
                               //

                              ooo.SelectObject(&b1);

                              CRect xy_(x*frmX-linB_FnP+ posX,  y*frmY-linB_FnP+ posY,
                                           x*frmX+linB_FnP+ posX,  y*frmY+linB_FnP+ posY);//xy
punkt

                              POINT xy_1;
                                       xy_1.x=x*frmX+linB_FnP+ posX;
                                       xy_1.y=y*frmY+linB_FnP+ posY;

                               if(swli==1)// erster wert
                               {
                                     ooo.MoveTo (x*frmX+ posX,  y*frmY+ posY );

                                     if(sw_xy==1) // pixel setzen x xy Punkt
                                     {
                                             ooo.SelectObject(&o2s1);

                                             if(mod_FnP==3)//kreuz
                                             {
                                                    ooo.MoveTo (xy_.left,xy_.top);
                                                    ooo.LineTo (xy_.right,xy_.bottom);
                                                    ooo.MoveTo (xy_.right,xy_.top);
                                                    ooo.LineTo (xy_.left,xy_.bottom);
                                             }

       if(mod_FnP==2)ooo.FrameRect(xy_,&b2);//ooo.FillSolidRect(xy_,fb_P)//eckig
                                             if(mod_FnP==1)ooo.RoundRect(xy_, xy_1); // rund
                                     }

                                     if(sw_x==1)//funktionswert x ausgeben
                                     {
                                             ooo.SelectObject(&ofx);
                                             ooo.SetTextColor(fn_x_fb);
                                             ooo.SetBkMode(TRANSPARENT);
                                             ooo.TextOut(x+ posX,  dlg.y-12+ posY,
ccx_);
                                     }

                                     if(sw_y==1)//funktionswert y ausgeben
                                     {
                                             ooo.SelectObject(&ofy);
                                             ooo.SetTextColor(fn_y_fb);
                                             ooo.SetBkMode(TRANSPARENT);
                                             ooo.TextOut(6+ posX,      y-12+ posY,
ccy_);
```

```
                                      }

                                       swli=0;// erster wert schalter

                                       //minima maxima
                                       x_m_min=x; y_m_min=y;
                                       x_m_max=x; y_m_max=y;

                                       yr_m_min=yr;   ys0_m_min=ys0; ysi0_m_min=ysi0;
yss0_m_min=yss0; yssi0_m_min=yssi0; ysr0_m_min=ysr0;ysri0_m_min=ysri0;
                                       yr_m_max=yr;   ys0_m_max=ys0; ysi0_m_max=ysi0;
yss0_m_max=yss0; yssi0_m_max=yssi0; ysr0_m_max=ysr0;ysri0_m_max=ysri0;
                                       yri_m_min=yri; ys1_m_min=ys1; ysi1_m_min=ysi1;
yss1_m_min=yss1; yssi1_m_min=yssi1; ysr1_m_min=ysr1;ysri1_m_min=ysri1;
                                       yri_m_max=yri; ys1_m_max=ys1; ysi1_m_max=ysi1;
yss1_m_max=yss1; yssi1_m_max=yssi1; ysr1_m_max=ysr1;ysri1_m_max=ysri1;
                               }

                               if(sw_FK==1)
                               if(swli==0) // folgende werte ----- Funktionskurve
                               {
                                      ooo.SelectObject(&o2s);
                                      if(sw_mod_==4)//linienunterbrechung bei F(e)
                                      {
                                            il++;
                                         if(il==(n_/2)+1){ooo.MoveTo (x*frmX+ posX,  y*frmY+
posY );}

                                         else {ooo.LineTo (x*frmX+ posX,  y*frmY+ posY );}
                                      }
                                      else
                                      {ooo.LineTo (x*frmX+ posX,  y*frmY+ posY );}
                               }

                               if(sw_xy==1)// pixel setzen x xy Punkte
                               {
                                      ooo.SelectObject(&o2s1);

                                      if(mod_FnP==3)//kreuz
                                      {
                                              ooo.MoveTo (xy_.left,xy_.top);
                                              ooo.LineTo (xy_.right,xy_.bottom);
                                              ooo.MoveTo (xy_.right,xy_.top);
                                              ooo.LineTo (xy_.left,xy_.bottom);
                                              ooo.MoveTo (x*frmX+ posX,  y*frmY+ posY );
                                      }

        if(mod_FnP==2)ooo.FrameRect(xy_,&b2);//ooo.FillSolidRect(xy_,fb_P)//eckig
                                      if(mod_FnP==1)ooo.RoundRect(xy_, xy_1); // rund
                               }

                               if(sw_x==1)//achsen beschriftung funktions werte
                               {
                                       ooo.SelectObject(&ofx);
                                       ooo.SetTextColor(fn_x_fb);
                                       ooo.SetBkMode(TRANSPARENT);

                                       ooo.TextOut(x*frmX +posFXx+ posX,  dlg.y*frmY +posFXy+
posY,    ccx_);
                               }

                               if(sw_y==1)
                               {
                                       ooo.SelectObject(&ofy);
                                       ooo.SetTextColor(fn_y_fb);
                                       ooo.SetBkMode(TRANSPARENT);

                                       ooo.TextOut(0*frmX +posFYx+ posX,  y*frmY +posFYy+ posY,
ccy_);
                               }

                               if(sw_xm==1) //achsen beschriftung funktions minmax
                               {
                                       ooo.SelectObject(&ofx);
                                       ooo.SetTextColor(fn_x_fb);
                                       ooo.SetBkMode(TRANSPARENT);
                                      if(x_==min_x+sc0x) ooo.TextOut(x*frmX +posFXx+ posX,
dlg.y*frmY +posFXy+ posY,    ccx_);
                                      if(x_==max_x+sc0x) ooo.TextOut(x*frmX +posFXx+ posX,
dlg.y*frmY +posFXy+ posY,    ccx_);
                               }

                               if(sw_ym==1)
                               {
                                      ooo.SelectObject(&ofy);
                                      ooo.SetTextColor(fn_y_fb);
                                      ooo.SetBkMode(TRANSPARENT);

                                      if(y_==min_y+sc0y)ooo.TextOut(0*frmX +posFYx+ posX,
y*frmY +posFYy+ posY,          ccy_);
                                      if(y_==max_y+sc0y)ooo.TextOut(0*frmX +posFYx+ posX,
y*frmY +posFYy+ posY,          ccy_);
                               }

                               //achsenskalen- und regressionsmarkierungs variablen

                               if(x> x_m_max) x_m_max=x; if(y> y_m_max) y_m_max=y;
                               if(x< x_m_min) x_m_min=x; if(y< y_m_min) y_m_min=y;

                               if(yr>  yr_m_max)  yr_m_max=yr;   if(ys0> ys0_m_max)
ys0_m_max=ys0; if(ysi0> ysi0_m_max) ysi0_m_max=ysi0;
                               if(yr<  yr_m_min)  yr_m_min=yr;   if(ys0< ys0_m_min)
ys0_m_min=ys0; if(ysi0< ysi0_m_min) ysi0_m_min=ysi0;
                               if(yri> yri_m_max) yri_m_max=yri; if(ys1> ys1_m_max)
ys1_m_max=ys1; if(ysi1> ysi1_m_max) ysi1_m_max=ysi1;
                               if(yri< yri_m_min) yri_m_min=yri; if(ys1< ys1_m_min)
ys1_m_min=ys1; if(ysi1< ysi1_m_min) ysi1_m_min=ysi1;

                               if(ysr0> ysr0_m_max) ysr0_m_max=ysr0;  if(ysr1> ysr1_m_max)
ysr1_m_max=ysr1;  if(yss0> yss0_m_max) yss0_m_max=yss0;
                               if(ysr0< ysr0_m_min) ysr0_m_min=ysr0;  if(ysr1< ysr1_m_min)
ysr1_m_min=ysr1;  if(yss0< yss0_m_min) yss0_m_min=yss0;
                               if(ysri0>ysri0_m_max)ysri0_m_max=ysri0;if(ysri1>ysri1_m_max)
ysri1_m_max=ysri1;if(yss1> yss1_m_max) yss1_m_max=yss1;
                               if(ysri0<ysri0_m_min)ysri0_m_min=ysri0;if(ysri1<ysri1_m_min)
ysri1_m_min=ysri1;if(yss1< yss1_m_min) yss1_m_min=yss1;

                               if(yssi0>yssi0_m_max) yssi0_m_max=yssi0;
                               if(yssi0<yssi0_m_min) yssi0_m_min=yssi0;
                               if(yssi1>yssi1_m_max) yssi1_m_max=yssi1;
                               if(yssi1<yssi1_m_min) yssi1_m_min=yssi1;

                      }while ( feof (f_) == 0); //
                                     fclose(f_);

                      int ni_=0;// über Funktionsvektoren

                      if(!filestr)do  //nicht filestream rendern
                      {

float fx_,            fy_;
                                                                      fx_=FVx_[ni_];
fy_=FVy_[ni_];
                                                                   x_=fx_+sc0x;
                                                                   y_=
fy_+sc0y;


                      /// einfügen
/////////////////////////////////////////////////////////////////////////////////////
                      //  |        // atof(cx_)=fx_, atof(cy_)=fy_
                      //  v

                                             char ccx_[20];
                                      sprintf(ccx_,"%s",      ftoc(x_-sc0x, ds_x));
//formatierte skalenwerte x
                              if(0)sprintf(ccx_,"%g", atof(ftoc(x_-sc0x, ds_x)));//zu
implementieren...

                                             char ccy_[20];
                                      sprintf(ccy_,"%s",      ftoc(y_-sc0y, ds_y));
//formatierte skalenwerte y
                              if(0)sprintf(ccx_,"%g", atof(ftoc(x_-sc0x, ds_x)));//zu
implementieren...
```

```
                              //koordinatenpunkt berechnung

                                             float pr_x = x_/(max_x+sc0x); //xi proportion
                                             float pr_y = y_/(max_y+sc0y); //yi proportion

                               x=           pr_x * dx_s  ; //dialog koordinatenpunkt zu wert x
                               y= dlg.y - ( pr_y * dy_s ); //dialog koordinatenpunkt zu wert y

                               x+=dx_sy;
                               y-=dy_sy;


////////////////////////////////////////////////////////////////////////////////////////////////
/////////////
                               // THETA rxy berechnung
                               //

                               if(sw_mod_==2) //Regressionsgeradenwert y
                               {
                                                                       float pr_yr;
                                                                       float pr_yri;
                                        if(sw_inv==0||rxy_D==1)pr_yr  = ((fx_* qR )
+sc0y)/(max_y+sc0y); //xir   proportion
                                        if(sw_inv==1||rxy_D==1)pr_yri = ((fy_* qR )
+sc0x)/(max_x+sc0x); //xir-1 proportion

                                       if(sw_inv==0||rxy_D==1)yr= dlg.y - ( pr_yr * dy_s );
//dialog koordinatenpunkt y'x
                                       if(sw_inv==0||rxy_D==1)yr-=dy_sy;

                                       if(sw_inv==1||rxy_D==1)yri= pr_yri * dx_s;
//dialog koordinatenpunkt x'y
                                       if(sw_inv==1||rxy_D==1)yri+=dx_sy;
                               }

                               if(sw_mod_==2)if(1) //Standardvorhersagefehler sy'x
                               {
                                                                            float pr_ys0;
                                                                            float pr_ys1;
                                                                            float pr_ysi0;
                                                                            float pr_ysi1;
                                        if(sw_inv==0||rxy_D==1) pr_ys0 =  (( (fx_* qR )-
qS*ci_zr) +sc0y)/(max_y+sc0y); //sy'- proportion
                                        if(sw_inv==0||rxy_D==1) pr_ys1 =  (( (fx_* qR
)+qS*ci_zr) +sc0y)/(max_y+sc0y); //sy'+ proportion
                                        if(sw_inv==1||rxy_D==1) pr_ysi0 = (( (fy_* qR )-
qS*ci_zr) +sc0x)/(max_x+sc0x); //sx'- proportion
                                        if(sw_inv==1||rxy_D==1) pr_ysi1 = (( (fy_* qR
)+qS*ci_zr) +sc0x)/(max_x+sc0x); //sx'+ proportion

                                       if(sw_inv==0||rxy_D==1)
                                       {
                                              ys0= dlg.y - ( pr_ys0 * dy_s ); //dialog
koordinatenpunkt sy'x-
                                              ys0-=dy_sy;

                                              ys1= dlg.y - ( pr_ys1 * dy_s ); //dialog
koordinatenpunkt sy'x+
                                              ys1-=dy_sy;
                                       }

                                       if(sw_inv==1||rxy_D==1)
                                       {
                                              ysi0=  pr_ysi0 * dx_s ;           //dialog
koordinatenpunkt sx'y-
                                              ysi0+=dx_sy;

                                              ysi1=  pr_ysi1 * dx_s ;           //dialog
koordinatenpunkt sx'y+
                                              ysi1+=dx_sy;
                                       }
                               }

                               if(sw_mod_==2)if(1) //Geschätzter Standardvorhersagefehler
s'y'x
                               {
```

```
                                                                          float pr_yss0;
                                                                          float pr_yss1;
                                                                          float pr_yssi0;
                                                                          float pr_yssi1;
                                       if(sw_inv==0||rxy_D==1) pr_yss0  = (( (fx_* qR )-
qsS*ci_zr) +sc0y)/(max_y+sc0y); //sy'- proportion
                                       if(sw_inv==0||rxy_D==1) pr_yss1  = (( (fx_* qR
)+qsS*ci_zr) +sc0y)/(max_y+sc0y); //sy'+ proportion
                                       if(sw_inv==1||rxy_D==1) pr_yssi0 = (( (fy_* qR )-
qsS*ci_zr) +sc0x)/(max_x+sc0x); //sx'- proportion
                                       if(sw_inv==1||rxy_D==1) pr_yssi1 = (( (fy_* qR
)+qsS*ci_zr) +sc0x)/(max_x+sc0x); //sx'+ proportion

                                      if(sw_inv==0||rxy_D==1)
                                      {
                                             yss0= dlg.y - ( pr_yss0 * dy_s ); //dialog
koordinatenpunkt s'y'x-
                                             yss0-=dy_sy;

                                             yss1= dlg.y - ( pr_yss1 * dy_s ); //dialog
koordinatenpunkt s'y'x+
                                             yss1-=dy_sy;
                                      }

                                      if(sw_inv==1||rxy_D==1)
                                      {
                                             yssi0=  pr_yssi0 * dx_s ;           //dialog
koordinatenpunkt s'x'y-
                                             yssi0+= dx_sy;

                                             yssi1=  pr_yssi1 * dx_s ;           //dialog
koordinatenpunkt s'x'y+
                                             yssi1+= dx_sy;
                                      }
                               }

                               if(sw_mod_==2)if(1) //Geschätzter Standardfehler der Regression
s'r
                               {
                                                                      float pr_ysr0;
                                                                      float pr_ysr1;
                                                                      float pr_ysri0;
                                                                      float pr_ysri1;
                                      if(sw_inv==0||rxy_D==1) pr_ysr0 = ((fx_* (qR-qsR*ci_tr
)) +sc0y)/(max_y+sc0y); //xir'   proportion
                                      if(sw_inv==0||rxy_D==1) pr_ysr1 = ((fx_* (qR+qsR*ci_tr
)) +sc0y)/(max_y+sc0y); //xir'   proportion
                                      if(sw_inv==1||rxy_D==1) pr_ysri0 = ((fy_* (qR-qsR*ci_tr
)) +sc0x)/(max_x+sc0x); //xir'-1 proportion
                                      if(sw_inv==1||rxy_D==1) pr_ysri1 = ((fy_* (qR+qsR*ci_tr
)) +sc0x)/(max_x+sc0x); //xir'-1 proportion

                                      if(sw_inv==0||rxy_D==1)
                                      {
                                             ysr0= dlg.y - ( pr_ysr0 * dy_s ); //dialog
koordinatenpunkt y'x r'
                                             ysr0-=dy_sy;

                                             ysr1= dlg.y - ( pr_ysr1 * dy_s );
                                             ysr1-=dy_sy;
                                      }

                                      if(sw_inv==1||rxy_D==1)
                                      {
                                             ysri0= pr_ysri0 * dx_s;             //dialog
koordinatenpunkt x'y r'
                                             ysri0+=dx_sy;

                                             ysri1= pr_ysri1 * dx_s;
                                             ysri1+=dx_sy;
                                      }
                               }


//////////////////////////////////////////////////////////////////////////////////////////////
/////////////
                               // Funktionszeichnung
                               //
```

```
                              ooo.SelectObject(&b1);

                              CRect xy_(x*frmX-linB_FnP+ posX,y*frmY-linB_FnP+ posY,
                                           x*frmX+linB_FnP+ posX,y*frmY+linB_FnP+ posY);//xy
punkt

                              POINT xy_1;
                                       xy_1.x=x*frmX+linB_FnP+ posX;
                                       xy_1.y=y*frmY+linB_FnP+ posY;

                               if(swli==1)
                               {
                                     ooo.MoveTo (x*frmX+ posX,  y*frmY+ posY ); // erster wert

                                     if(sw_xy==1) // pixel setzen x xy Punkt
                                     {
                                            ooo.SelectObject(&o2s1);

                                            if(mod_FnP==3)//kreuz
                                            {
                                                   ooo.MoveTo (xy_.left,xy_.top);
                                                   ooo.LineTo (xy_.right,xy_.bottom);
                                                   ooo.MoveTo (xy_.right,xy_.top);
                                                   ooo.LineTo (xy_.left,xy_.bottom);
                                            }

       if(mod_FnP==2)ooo.FrameRect(xy_,&b2);//ooo.FillSolidRect(xy_,fb_P)//eckig
                                            if(mod_FnP==1)ooo.RoundRect(xy_, xy_1); // rund
                                     }

                                     if(sw_x==1)//funktionswert x ausgeben
                                     {
                                            ooo.SelectObject(&ofx);
                                            ooo.SetTextColor(fn_x_fb);
                                            ooo.SetBkMode(TRANSPARENT);
                                            ooo.TextOut(x*frmX+ posX,  dlg.y*frmY-12+ posY,
ccx_);
                                     }

                                     if(sw_y==1)//funktionswert y ausgeben
                                     {
                                            ooo.SelectObject(&ofy);
                                            ooo.SetTextColor(fn_y_fb);
                                            ooo.SetBkMode(TRANSPARENT);
                                            ooo.TextOut(6*frmX+ posX,      y*frmY-12+ posY,
ccy_);
                                     }

                                      swli=0;//

                                      //minima maxima
                                      x_m_min=x; y_m_min=y;
                                      x_m_max=x; y_m_max=y;

                                      yr_m_min=yr;   ys0_m_min=ys0; ysi0_m_min=ysi0;
yss0_m_min=yss0; yssi0_m_min=yssi0; ysr0_m_min=ysr0;ysri0_m_min=ysri0;
                                      yr_m_max=yr;   ys0_m_max=ys0; ysi0_m_max=ysi0;
yss0_m_max=yss0; yssi0_m_max=yssi0; ysr0_m_max=ysr0;ysri0_m_max=ysri0;
                                      yri_m_min=yri; ys1_m_min=ys1; ysi1_m_min=ysi1;
yss1_m_min=yss1; yssi1_m_min=yssi1; ysr1_m_min=ysr1;ysri1_m_min=ysri1;
                                      yri_m_max=yri; ys1_m_max=ys1; ysi1_m_max=ysi1;
yss1_m_max=yss1; yssi1_m_max=yssi1; ysr1_m_max=ysr1;ysri1_m_max=ysri1;
                               }

                               if(sw_FK==1)
                               if(swli==0) // folgende werte ----- Funktionskurve
                               {
                                     ooo.SelectObject(&o2s);
                                     if(sw_mod_==4)//linienunterbrechung bei F(e)
                                     {
                                           il++;
                                        if(il==(n_/2)+1){ooo.MoveTo (x*frmX+ posX,  y*frmY+
posY );}

                                        else {ooo.LineTo (x*frmX+ posX,  y*frmY+ posY );}
                                     }
                                     else
```

```
                                      {ooo.LineTo (x*frmX+ posX,  y*frmY+ posY );}
                               }

                               if(sw_xy==1)// pixel setzen x xy Punkte
                               {
                                      ooo.SelectObject(&o2s1);

                                      if(mod_FnP==3)//kreuz
                                      {
                                              ooo.MoveTo (xy_.left,xy_.top);
                                              ooo.LineTo (xy_.right,xy_.bottom);
                                              ooo.MoveTo (xy_.right,xy_.top);
                                              ooo.LineTo (xy_.left,xy_.bottom);
                                              ooo.MoveTo (x+ posX,  y );
                                      }

        if(mod_FnP==2)ooo.FrameRect(xy_,&b2);//ooo.FillSolidRect(xy_,fb_P)//eckig
                                      if(mod_FnP==1)ooo.RoundRect(xy_, xy_1); // rund
                               }



                               if(sw_x==1)//achsen beschriftung funktions werte
                               {
                                       ooo.SelectObject(&ofx);
                                       ooo.SetTextColor(fn_x_fb);
                                       ooo.SetBkMode(TRANSPARENT);

                                       ooo.TextOut(x*frmX +posFXx+ posX,  dlg.y*frmY +posFXy+
posY,    ccx_);
                               }

                               if(sw_y==1)
                               {
                                       ooo.SelectObject(&ofy);
                                       ooo.SetTextColor(fn_y_fb);
                                       ooo.SetBkMode(TRANSPARENT);

                                       ooo.TextOut(0*frmX +posFYx+ posX,  y*frmY +posFYy+ posY,
ccy_);
                               }

                               if(sw_xm==1) //achsen beschriftung funktions minmax
                               {
                                       ooo.SelectObject(&ofx);
                                       ooo.SetTextColor(fn_x_fb);
                                       ooo.SetBkMode(TRANSPARENT);

                                       if(x_==min_x+sc0x) ooo.TextOut(x*frmX +posFXx+ posX,
dlg.y*frmY +posFXy+ posY,    ccx_);
                                       if(x_==max_x+sc0x) ooo.TextOut(x*frmX +posFXx+ posX,
dlg.y*frmY +posFXy+ posY,    ccx_);
                               }

                               if(sw_ym==1)
                               {
                                       ooo.SelectObject(&ofy);
                                       ooo.SetTextColor(fn_y_fb);
                                       ooo.SetBkMode(TRANSPARENT);

                                       if(y_==min_y+sc0y)ooo.TextOut(0*frmX +posFYx+ posX,
y*frmY +posFYy+ posY,          ccy_);
                                       if(y_==max_y+sc0y)ooo.TextOut(0*frmX +posFYx+ posX,
y*frmY +posFYy+ posY,          ccy_);
                               }

                               //achsenskalen- und regressionsmarkierungs variablen

                               if(x> x_m_max) x_m_max=x; if(y> y_m_max) y_m_max=y;
                               if(x< x_m_min) x_m_min=x; if(y< y_m_min) y_m_min=y;

                               if(yr>  yr_m_max)  yr_m_max=yr;   if(ys0> ys0_m_max)
ys0_m_max=ys0; if(ysi0> ysi0_m_max) ysi0_m_max=ysi0;
                               if(yr<  yr_m_min)  yr_m_min=yr;   if(ys0< ys0_m_min)
ys0_m_min=ys0; if(ysi0< ysi0_m_min) ysi0_m_min=ysi0;
                               if(yri> yri_m_max) yri_m_max=yri; if(ys1> ys1_m_max)
ys1_m_max=ys1; if(ysi1> ysi1_m_max) ysi1_m_max=ysi1;
```

```
                              if(yri< yri_m_min) yri_m_min=yri; if(ys1< ys1_m_min)
ys1_m_min=ys1; if(ysi1< ysi1_m_min) ysi1_m_min=ysi1;

                              if(ysr0> ysr0_m_max) ysr0_m_max=ysr0;  if(ysr1> ysr1_m_max)
ysr1_m_max=ysr1;  if(yss0> yss0_m_max) yss0_m_max=yss0;
                              if(ysr0< ysr0_m_min) ysr0_m_min=ysr0;  if(ysr1< ysr1_m_min)
ysr1_m_min=ysr1;  if(yss0< yss0_m_min) yss0_m_min=yss0;
                              if(ysri0>ysri0_m_max)ysri0_m_max=ysri0;if(ysri1>ysri1_m_max)
ysri1_m_max=ysri1;if(yss1> yss1_m_max) yss1_m_max=yss1;
                              if(ysri0<ysri0_m_min)ysri0_m_min=ysri0;if(ysri1<ysri1_m_min)
ysri1_m_min=ysri1;if(yss1< yss1_m_min) yss1_m_min=yss1;

                              if(yssi0>yssi0_m_max) yssi0_m_max=yssi0;
                              if(yssi0<yssi0_m_min) yssi0_m_min=yssi0;
                              if(yssi1>yssi1_m_max) yssi1_m_max=yssi1;
                              if(yssi1<yssi1_m_min) yssi1_m_min=yssi1;

                      //  ^
                      //  |
                      /// einfügen
//////////////////////////////////////////////////////////////////////////////////////////////

                                ni_++;
                      }while(ni_<n_); //Funktionsvektoren


                      swli=1; ///


       ///////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////
                      // Funktionswertschleife ende
///////////////////////////////////////////////////////////////////////////////////

       ///////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////



       ///////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////
                      // THETA rxy  Achsen zeichnen
                      //

                      if(sw_mod_==2) // Regressionsgerade
                      {
                             ooo.SelectObject(&of1); //font y-achse

                                                                CString cr_;
                             if(sw_inv==0||rxy_D==1)if(rxy_==1)
                             {
                                    ooo.SetTextColor(fb_r); //textfarbe
                                    ooo.SelectObject(&or1);                cr_="r(xy)";
                                                                 if(r_q==1)cr_+="= ";

if(r_q==1)cr_+=ftoc(qR,3);
                                    if(qR>=0)
                                    {
                                                  ooo.MoveTo (x_m_min*frmX+ posX,
yr_m_max*frmY+ posY );
                                                  ooo.LineTo (x_m_max*frmX+ posX,
yr_m_min*frmY+ posY );
                                           if(r_q)ooo.TextOut(x_m_max*frmX+3+
posX,yr_m_min*frmY-5+ posY,  cr_);//regressionswert rxy
                                    }

                                    if(qR<0)
                                    {
                                                  ooo.MoveTo (x_m_min*frmX+ posX,
yr_m_min*frmY+ posY );
                                                  ooo.LineTo (x_m_max*frmX+ posX,
yr_m_max*frmY+ posY );
                                           if(r_q)ooo.TextOut(x_m_max*frmX+3+
posX,yr_m_max*frmY-5+ posY,  cr_);//regressionswert rxy
                                    }
                             }

                             if(sw_inv==1||rxy_D==1)if(ryx_==1)
```

```
                              {
                                     ooo.SetTextColor(fb_ri); //textfarbe
                                     ooo.SelectObject(&or1i);                cr_="r(yx)";
                                                                  if(ri_q==1)cr_+="= ";
                                                                  if(ri_q==1)cr_+=ftoc(qR,3);
                                     if(qR>=0)
                                     {
                                                    ooo.MoveTo (yri_m_max*frmX+ posX,
y_m_min*frmY+ posY);
                                                    ooo.LineTo (yri_m_min*frmX+ posX,
y_m_max*frmY+ posY);
                                            if(ri_q)ooo.TextOut(yri_m_min*frmX+3+
posX,y_m_max*frmY-5+ posY,  cr_);//regressionswert ryx
                                     }

                                     if(qR<0)
                                     {
                                                    ooo.MoveTo (yri_m_max*frmX+ posX,
y_m_max*frmY+ posY);
                                                    ooo.LineTo (yri_m_min*frmX+ posX,
y_m_min*frmY+ posY);
                                            if(ri_q)ooo.TextOut(yri_m_min*frmX+3+
posX,y_m_min*frmY-5+ posY,  cr_);//regressionswert ryx
                                     }
                              }
                      }

                      if(sw_mod_==2) // Standardvorhersagefehler
                      {
                              ooo.SelectObject(&of1); //font y-achse

                              if(sw_inv==0||rxy_D==1)if(syx_==1)
                              {
                                     ooo.SetTextColor(fb_syx); //textfarbe
                                     ooo.SelectObject(&or4);                CString
cr_="□·sy'x";
                                                                        if(syx_q==1)cr_+="= ";

          if(syx_q==1)cr_+=ftoc(qS*ci_zr,3);
                                     if(qR>=0)
                                     {
                                                     ooo.MoveTo (x_m_min*frmX+ posX,
ys0_m_max*frmY+ posY );
                                                     ooo.LineTo (x_m_max*frmX+ posX,
ys0_m_min*frmY+ posY );
                                            if(syx_q)ooo.TextOut(x_m_max*frmX+3+
posX,ys0_m_min*frmY-5+ posY,  cr_);// sy'x wert
                                                     ooo.MoveTo (x_m_min*frmX+ posX,
ys1_m_max*frmY+ posY );
                                                     ooo.LineTo (x_m_max*frmX+ posX,
ys1_m_min*frmY+ posY );
                                            if(syx_q)ooo.TextOut(x_m_max*frmX+3+
posX,ys1_m_min*frmY-5+ posY,  cr_);// sy'x wert
                                     }

                                     if(qR<0)
                                     {
                                                     ooo.MoveTo (x_m_min*frmX+ posX,
ys0_m_min*frmY+ posY );
                                                     ooo.LineTo (x_m_max*frmX+ posX,
ys0_m_max*frmY+ posY );
                                            if(syx_q)ooo.TextOut(x_m_max*frmX+3+
posX,ys0_m_max*frmY-5+ posY,  cr_);// sy'x wert
                                                     ooo.MoveTo (x_m_min*frmX+ posX,
ys1_m_min*frmY+ posY );
                                                     ooo.LineTo (x_m_max*frmX+ posX,
ys1_m_max*frmY+ posY );
                                            if(syx_q)ooo.TextOut(x_m_max*frmX+3+
posX,ys1_m_max*frmY-5+ posY,  cr_);// sy'x wert

                                     }
                              }
                              if(sw_inv==1||rxy_D==1)if(sxy_==1)
                              {
                                     ooo.SetTextColor(fb_sxy); //textfarbe
                                     ooo.SelectObject(&or4i);               CString
cr_="□·sx'(y)";
                                                                        if(sxy_q==1)cr_+="= ";
          if(sxy_q==1)cr_+=ftoc(qS*ci_zr,3);
                                     if(qR>=0)
                                     {
                                                     ooo.MoveTo (ysi0_m_max*frmX+
posX,y_m_min*frmY+ posY );
                                                     ooo.LineTo (ysi0_m_min*frmX+
posX,y_m_max*frmY+ posY );
                                            if(sxy_q)ooo.TextOut(ysi0_m_min*frmX+3+
posX,y_m_max*frmY-5+ posY,  cr_);// sx'y wert
                                                     ooo.MoveTo (ysi1_m_max*frmX+
posX,y_m_min*frmY+ posY );
                                                     ooo.LineTo (ysi1_m_min*frmX+
posX,y_m_max*frmY+ posY );
                                            if(sxy_q)ooo.TextOut(ysi1_m_min*frmX+3+
posX,y_m_max*frmY-5+ posY,  cr_);// sx'y wert

                                     }
                                     if(qR<0)
                                     {
                                                     ooo.MoveTo (ysi0_m_max*frmX+
posX,y_m_max*frmY+ posY );
                                                     ooo.LineTo (ysi0_m_min*frmX+
posX,y_m_min*frmY+ posY );
                                            if(sxy_q)ooo.TextOut(ysi0_m_min*frmX+3+
posX,y_m_min*frmY-5+ posY,  cr_);// sx'y wert
                                                            ooo.MoveTo (ysi1_m_max*frmX+
posX,y_m_max*frmY+ posY );
                                                     ooo.LineTo (ysi1_m_min*frmX+
posX,y_m_min*frmY+ posY );
                                            if(sxy_q)ooo.TextOut(ysi1_m_min*frmX+3+
posX,y_m_min*frmY-5+ posY,  cr_);// sx'y wert
                                     }
                              }
                      }

                      if(sw_mod_==2) // Geschätzter Standardvorhersagefehler
                      {
                              ooo.SelectObject(&of1); //font y-achse

                              if(sw_inv==0||rxy_D==1)if(s1yx_==1)
                              {
                                     ooo.SetTextColor(fb_sgyx); //textfarbe
                                     ooo.SelectObject(&or5);                CString
cr_="□·ôy'x";
                                                                       if(sgyx_q==1)cr_+="= ";

         if(sgyx_q==1)cr_+=ftoc(qsS*ci_zr,3);

                                     if(qR>=0)
                                     {
                                                      ooo.MoveTo (x_m_min*frmX+ posX,
yss0_m_max*frmY+ posY );
                                                      ooo.LineTo (x_m_max*frmX+ posX,
yss0_m_min*frmY+ posY );
                                            if(sgyx_q)ooo.TextOut(x_m_max*frmX+3+
posX,yss0_m_min*frmY-5+ posY,  cr_);// ôy'x wert
                                                      ooo.MoveTo (x_m_min*frmX+ posX,
yss1_m_max*frmY+ posY );
                                                      ooo.LineTo (x_m_max*frmX+ posX,
yss1_m_min*frmY+ posY );
                                            if(sgyx_q)ooo.TextOut(x_m_max*frmX+3+
posX,yss1_m_min*frmY-5+ posY,  cr_);// ôy'x wert
                                     }

                                     if(qR<0)
                                     {
                                                      ooo.MoveTo (x_m_min*frmX+ posX,
yss0_m_min*frmY+ posY );
                                                      ooo.LineTo (x_m_max*frmX+ posX,
yss0_m_max*frmY+ posY );
                                            if(sgyx_q)ooo.TextOut(x_m_max*frmX+3+
posX,yss0_m_max*frmY-5+ posY,  cr_);// ôy'x wert
                                                      ooo.MoveTo (x_m_min*frmX+ posX,
yss1_m_min*frmY+ posY );
                                                      ooo.LineTo (x_m_max*frmX+ posX,
yss1_m_max*frmY+ posY );
```

```
                                            if(sgyx_q)ooo.TextOut(x_m_max*frmX+3+
posX,yss1_m_max*frmY-5+ posY,  cr_);// ôy'x wert
                                     }
                              }
                              if(sw_inv==1||rxy_D==1)if(s1xy_==1)
                              {
                                     ooo.SetTextColor(fb_sgxy); //textfarbe
                                     ooo.SelectObject(&or5i);                CString
cr_="□·ôx'y";
                                                                        if(sgxy_q==1)cr_+="=
";

          if(sgxy_q==1)cr_+=ftoc(qsS*ci_zr,3);

                                     if(qR>=0)
                                     {
                                                      ooo.MoveTo (yssi0_m_max*frmX+
posX,y_m_min*frmY+ posY );
                                                      ooo.LineTo (yssi0_m_min*frmX+
posX,y_m_max*frmY+ posY );
                                            if(sgxy_q)ooo.TextOut(yssi0_m_min*frmX+3+
posX,y_m_max*frmY-5+ posY,  cr_);// ôx'y wert
                                                      ooo.MoveTo (yssi1_m_max*frmX+
posX,y_m_min*frmY+ posY );
                                                      ooo.LineTo (yssi1_m_min*frmX+
posX,y_m_max*frmY+ posY );
                                            if(sgxy_q)ooo.TextOut(yssi1_m_min*frmX+3+
posX,y_m_max*frmY-5+ posY,  cr_);// ôx'y wert
                                     }

                                     if(qR<0)
                                     {
                                                      ooo.MoveTo (yssi0_m_max*frmX+
posX,y_m_max*frmY+ posY );
                                                      ooo.LineTo (yssi0_m_min*frmX+
posX,y_m_min*frmY+ posY );
                                            if(sgxy_q)ooo.TextOut(yssi0_m_min*frmX+3+
posX,y_m_min*frmY-5+ posY,  cr_);// ôx'y wert
                                                      ooo.MoveTo (yssi1_m_max*frmX+
posX,y_m_max*frmY+ posY );
                                                      ooo.LineTo (yssi1_m_min*frmX+
posX,y_m_min*frmY+ posY );
                                            if(sgxy_q)ooo.TextOut(yssi1_m_min*frmX+3+
posX,y_m_min*frmY-5+ posY,  cr_);// ôx'y wert
                                     }
                              }
                       }

                       if(sw_mod_==2) // Geschätzter Korrelationsfehler
                       {
                              ooo.SelectObject(&of1); //font y-achse

                              if(sw_inv==0||rxy_D==1)if(srxy_==1)
                              {
                                     ooo.SetTextColor(fb_sr); //textfarbe
                                     ooo.SelectObject(&or2);                CString
cr_="t·ôr(xy)";
                                                                         if(sr_q==1)cr_+="= ";

if(sr_q==1)cr_+=ftoc(qsR*ci_tr,3);
                                     if(qR>=0)
                                     {
                                                    ooo.MoveTo (x_m_min*frmX+ posX,
ysr0_m_max*frmY+ posY );
                                                    ooo.LineTo (x_m_max*frmX+ posX,
ysr0_m_min*frmY+ posY );
                                            if(sr_q)ooo.TextOut(x_m_max*frmX+3+
posX,ysr0_m_min*frmY-5+ posY,  cr_);// ôrxy wert
                                                    ooo.MoveTo (x_m_min*frmX+ posX,
ysr1_m_max*frmY+ posY );
                                                    ooo.LineTo (x_m_max*frmX+ posX,
ysr1_m_min*frmY+ posY );
                                     }

                                     if(qR<0)
                                     {
                                                    ooo.MoveTo (x_m_min*frmX+ posX,
ysr0_m_min*frmY+ posY );
```

```
                                                     ooo.LineTo (x_m_max*frmX+ posX,
ysr0_m_max*frmY+ posY );
                                             if(sr_q)ooo.TextOut(x_m_max*frmX+3+
posX,ysr0_m_max*frmY-5+ posY,  cr_);// ôrxy wert
                                                     ooo.MoveTo (x_m_min*frmX+ posX,
ysr1_m_min*frmY+ posY );
                                                     ooo.LineTo (x_m_max*frmX+ posX,
ysr1_m_max*frmY+ posY );
                                     }
                              }

                              if(sw_inv==1||rxy_D==1)if(sryx_==1)
                              {
                                     ooo.SetTextColor(fb_sri); //textfarbe
                                     ooo.SelectObject(&or2i);               CString
cr_="t·ôr(yx)";
                                                                         if(sri_q==1)cr_+="=
";

if(sri_q==1)cr_+=ftoc(qsR*ci_tr,3);
                                     if(qR>=0)
                                     {
                                                      ooo.MoveTo (ysri0_m_max*frmX+ posX,
y_m_min*frmY+ posY);
                                                      ooo.LineTo (ysri0_m_min*frmX+ posX,
y_m_max*frmY+ posY);
                                             if(sri_q)ooo.TextOut(ysri0_m_min*frmX+3+
posX,y_m_max*frmY-5+ posY,  cr_);// ôryx wert
                                                      ooo.MoveTo (ysri1_m_max*frmX+ posX,
y_m_min*frmY+ posY);
                                                      ooo.LineTo (ysri1_m_min*frmX+ posX,
y_m_max*frmY+ posY);
                                     }
                                     if(qR<0)
                                     {
                                                      ooo.MoveTo (ysri0_m_max*frmX+ posX,
y_m_max*frmY+ posY);
                                                      ooo.LineTo (ysri0_m_min*frmX+ posX,
y_m_min*frmY+ posY);
                                             if(sri_q)ooo.TextOut(ysri0_m_min*frmX+3+
posX,y_m_min*frmY-5+ posY,  cr_);// ôryx wert
                                                      ooo.MoveTo (ysri1_m_max*frmX+ posX,
y_m_max*frmY+ posY);
                                                      ooo.LineTo (ysri1_m_min*frmX+ posX,
y_m_min*frmY+ posY);
                                     }
                              }
                      }

                      if(sw_mod_==2) // Geschätzter Regressionsfehler
                      {
                              float tmp_0=0;
                              float tmp_1=0;

                              if(sw_inv==0||rxy_D==1)if(srx_==1)
                              {
                                     ooo.SelectObject(&of1); //font y-achse
                                     ooo.SetTextColor(fb_sR); //textfarbe
                                     ooo.SelectObject(&or3);

                                     float dx= ((max_x-min_x) /100);
                                     float dx1=0;
                                     float x_m_1=((x_m_max-x_m_min)/(100));

                                     for(float x_m= x_m_min ;x_m<=x_m_max;x_m+=x_m_1 )
                                     {
                                             float pr_ysp0 = ((( (min_x+dx1)*qR )-(qsS
*sqrt(1/n_+pow(min_x+dx1,2)/n_))*ci_tr) +sc0y)/(max_y+sc0y); //s'Y'- proportion
                                             float pr_ysp1 = ((( (min_x+dx1)*qR )+(qsS
*sqrt(1/n_+pow(min_x+dx1,2)/n_))*ci_tr) +sc0y)/(max_y+sc0y); //s'Y'+ proportion

dx1+=dx;
                                                      pr_ysp0= dlg.y - ( pr_ysp0 * dy_s );
//dialog koordinatenpunkt s'Y'x-
                                                      pr_ysp0-=dy_sy;

                                                      pr_ysp1= dlg.y - ( pr_ysp1 * dy_s );
//dialog koordinatenpunkt s'Y'x+
                                                      pr_ysp1-=dy_sy;
                                             if(x_m>x_m_min)
                                             {
                                                    ooo.MoveTo (x_m*frmX-x_m_1+ posX, 
tmp_0*frmY+ posY  );
                                                    ooo.LineTo (x_m*frmX + posX     , 
pr_ysp0*frmY+ posY);


                                                    ooo.MoveTo (x_m*frmX-x_m_1+ posX, 
tmp_1*frmY+ posY  );
                                                    ooo.LineTo (x_m*frmX  + posX    , 
pr_ysp1*frmY+ posY);
                                              if(sR_q)if(x_m>=x_m_max-x_m_1)
                                                    ooo.TextOut(x_m*frmX+3+ posX,pr_ysp1*frmY-
5+ posY,  "t·ôR(x)");// ôRx
                                             }

tmp_0=pr_ysp0;

tmp_1=pr_ysp1;
                                     }
                              }

                              if(sw_inv==1||rxy_D==1)if(sry_==1)
                              {
                                     ooo.SelectObject(&of1); //font y-achse
                                     ooo.SetTextColor(fb_sRi); //textfarbe
                                     ooo.SelectObject(&or3i);

                                     float dy= ((max_y-min_y) /(100));
                                     float dy1=0;
                                     float y_m_1=((y_m_max-y_m_min)/(100));

                                     for(float y_m= y_m_max ;y_m>=y_m_min;y_m-=y_m_1 )
                                     {
                                             float pr_ysp0 = ((( (min_y+dy1)*qR )-(qsS
*sqrt(1/n_+pow(min_y+dy1,2)/n_))*ci_tr) +sc0x)/(max_x+sc0x); //s'X'- proportion
                                             float pr_ysp1 = ((( (min_y+dy1)*qR )+(qsS
*sqrt(1/n_+pow(min_y+dy1,2)/n_))*ci_tr) +sc0x)/(max_x+sc0x); //s'X'+ proportion

dy1+=dy;

                                                      pr_ysp0= pr_ysp0 * dx_s ; //dialog 
koordinatenpunkt s'X'y-
                                                      pr_ysp0+=dx_sy;
                                                      pr_ysp1= pr_ysp1 * dx_s ; //dialog 
koordinatenpunkt s'X'y+
                                                      pr_ysp1+=dx_sy;

                                             if(y_m<y_m_max)
                                             {
                                                    ooo.MoveTo (tmp_0*frmX+ posX, 
y_m*frmY+y_m_1+ posY);
                                                    ooo.LineTo (pr_ysp0*frmX+ posX, y_m*frmY+ 
posY     );


                                                    ooo.MoveTo (tmp_1*frmX+ posX, 
y_m*frmY+y_m_1+ posY);
                                                    ooo.LineTo (pr_ysp1*frmX+ posX, y_m*frmY + 
posY     );
                                                    if(sRi_q)if(y_m<=y_m_min+y_m_1)
                                                    ooo.TextOut(pr_ysp1*frmX+3+ posX,y_m*frmY-
5+ posY,  "t·ôR(y)");// ôRy
                                             }
                                                              tmp_0=pr_ysp0;
                                                              tmp_1=pr_ysp1;
                                     }
                              }
                       }


       ////////////////////////////////////////////////////////////////////////////////
//////////////////////////
                       // THETA Fp Achsen zeichnen
                       //

                       if(sw_mod_==3)if(am_) // F(p) am linie
                       {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_am); //textfarbe
                                                           CString cr_="x";
                              ooo.SelectObject(&op1);          if(am_q==1)cr_+="=";
                                                          if(am_q==1)cr_+=ftoc(qY.am,3);

                              if(sw_inv==0) //F(p)
                              {
                                     float pr_yam  = (qY.am +sc0y)/(max_y+sc0y); //qam
proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qam y
                                              pr_yam-=dy_sy;

                                             ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                             ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(am_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  am wert y
                              }

                              if(sw_inv==1)//F-1(p)
                              {
                                     float pr_xam  = (qY.am +sc0x)/(max_x+sc0x); //qam
proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qam x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(am_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-10+
posY,  cr_);//  am wert x
                              }
                       }

                       if(sw_mod_==3)if(sd_) // F(p) sd linie
                       {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_sd); //textfarbe
                                                                CString cr_="□·s";
                              ooo.SelectObject(&op2);          if(sd_q==1)cr_+="= ";
                                                          if(sd_q==1)cr_+=ftoc(qY.sd*ci_zp,3);

                              if(sw_inv==0) //F(p)
                              {
                                     float pr_yam  = (qY.am+(qY.sd)*ci_zp +sc0y)/(max_y+sc0y);
//qsd   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qsd+ y
                                              pr_yam-=dy_sy;

                                             ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                             ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(sd_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  sd wert y

                                              pr_yam  = (qY.am-(qY.sd)*ci_zp
+sc0y)/(max_y+sc0y); //qsd   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qsd- y
                                              pr_yam-=dy_sy;

                                             ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                             ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(sd_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  sd wert y
                              }

                              if(sw_inv==1)//F-1(p)
                              {
                                     float pr_xam  = (qY.am+(qY.sd)*ci_zp +sc0x)/(max_x+sc0x);
//qsd   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qsd+ x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(sd_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-10+
posY,  cr_);//  sd wert x

                                              pr_xam  = (qY.am-(qY.sd)*ci_zp
+sc0x)/(max_x+sc0x); //qsd   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qsd- x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(sd_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-10+
posY,  cr_);//  sd wert x
                              }
                      }

                      if(sw_mod_==3)if(sdg_) // F(p) sd' linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_sdg); //textfarbe
                                                                  CString cr_="□·ô";
                              ooo.SelectObject(&op4);          if(sdg_q==1)cr_+="= ";

if(sdg_q==1)cr_+=ftoc(qY.sdg*ci_zp,3);

                              if(sw_inv==0) //F(p)
                              {
                                     float pr_yam  = (qY.am+(qY.sdg)*ci_zp
+sc0y)/(max_y+sc0y); //qsd'   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qsd'+ y
                                              pr_yam-=dy_sy;

                                              ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                              ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(sdg_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  ô wert y

                                              pr_yam  = (qY.am-(qY.sdg)*ci_zp
+sc0y)/(max_y+sc0y); //qsd'   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qsd'- y
                                              pr_yam-=dy_sy;

                                              ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                              ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(sdg_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  ô wert y
                              }

                              if(sw_inv==1)//F-1(p)
                              {
                                     float pr_xam  = (qY.am+(qY.sdg)*ci_zp
+sc0x)/(max_x+sc0x); //qsd'   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qsd+ x
                                              pr_xam+=dx_sy;

                                              ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                              ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
```

```
                                     if(sdg_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-10+
posY,  cr_);//  ô wert x

                                              pr_xam  = (qY.am-(qY.sdg)*ci_zp
+sc0x)/(max_x+sc0x); //qsd'   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qsd'- x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(sdg_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-10+
posY,  cr_);//  ô wert x
                              }
                      }

                      if(sw_mod_==3)if(sgam_) // F(p) s'am linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_sgam); //textfarbe
                                                                  CString cr_="t·ôx";
                              ooo.SelectObject(&op3);          if(sgam_q==1)cr_+="= ";

if(sgam_q==1)cr_+=ftoc(qY.sgam*ci_tp,3);
                              if(sw_inv==0) //F(p)
                              {
                                     float pr_yam  = (qY.am+(qY.sgam)*ci_tp
+sc0y)/(max_y+sc0y); //qs'am   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qs'am + y
                                              pr_yam-=dy_sy;

                                               ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                                      ooo.LineTo (x_m_max*frmX+ posX,
pr_yam*frmY+ posY);

                                              pr_yam  = (qY.am-(qY.sgam)*ci_tp
+sc0y)/(max_y+sc0y); //qs'am   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qs'am - y
                                              pr_yam-=dy_sy;

                                               ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                               ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(sgam_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  ôx wert y
                              }

                              if(sw_inv==1)//F-1(p)
                              {
                                     float pr_xam  = (qY.am+(qY.sgam)*ci_tp
+sc0x)/(max_x+sc0x); //qs'am   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qs'am+ x
                                              pr_xam+=dx_sy;

                                               ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                               ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);

                                              pr_xam  = (qY.am-(qY.sgam)*ci_tp
+sc0x)/(max_x+sc0x); //qs'am   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qs'am - x
                                              pr_xam+=dx_sy;

                                               ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                               ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(sgam_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-
10+ posY,  cr_);//  ôx wert x
```

```
                              }
                      }

                      if(sw_mod_==3)if(a3_) // F(p) a3 linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_a3); //textfarbe
                                                              CString cr_="a3";
                              ooo.SelectObject(&op5);       if( a3_q==1)cr_+="= ";
                                                       if( a3_q==1)cr_+=ftoc(qY.a3,3);
                                                       if( a3_q==1)cr_+="z";

                              if(sw_inv==0) //F(p)
                              {
                                     float pr_yam  = ((qY.am+qY.a3*qY.sd) +sc0y)/(max_y+sc0y);
//qa3   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qa3 y
                                              pr_yam-=dy_sy;

                                             ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                             ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(a3_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  a3 wert y
                              }

                              if(sw_inv==1)//F-1(p)
                              {
                                     float pr_xam  = ((qY.am+qY.a3*qY.sd) +sc0x)/(max_x+sc0x);
//qa3   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qa3 x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(a3_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-19+
posY,  cr_);//  a3 wert x
                              }
                      }

                      if(sw_mod_==3)if(a4_) // F(p) a4 linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_a4); //textfarbe
                                                                CString cr_="a4";
                              ooo.SelectObject(&op8);          if(a4_q==1)cr_+="= ";
                                                          if(a4_q==1)cr_+=ftoc(qY.a4,3);
                                                          if(a4_q==1)cr_+="z";

                              if(sw_inv==0) //F(p)
                              {
                                     float pr_yam  = ((qY.am+qY.a4*qY.sd) +sc0y)/(max_y+sc0y);
//qa4   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qa4+ y
                                              pr_yam-=dy_sy;

                                             ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                             ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);

                                              pr_yam  = ((qY.am-qY.a4*qY.sd)
+sc0y)/(max_y+sc0y); //qa4   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qa4- y
                                              pr_yam-=dy_sy;

                                             ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                             ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
```

```
                                     if(a4_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  a4 wert y
                              }

                              if(sw_inv==1)//F-1(p)
                              {
                                     float pr_xam  = ((qY.am+qY.a4*qY.sd) +sc0x)/(max_x+sc0x);
//qa4   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qa4+ x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);

                                              pr_xam  = ((qY.am-qY.a4*qY.sd)
+sc0x)/(max_x+sc0x); //qa4   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qa4- x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(a4_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-19+
posY,  cr_);//  a4 wert x
                              }
                      }

                      if(sw_mod_==3)if(sga3_) // F(p) sga3 linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_sga3); //textfarbe
                                                                 CString cr_="t·ôa3";
                              ooo.SelectObject(&op7);          if(sa3g_q==1)cr_+="= ";

if(sa3g_q==1)cr_+=ftoc(qY.sga3*ci_tp,3);
                                                          if(sa3g_q==1)cr_+="z";

                              if(sw_inv==0) //F(p)
                              {
                                     //float pr_yam  = ((qY.am+(qY.sga3*ci_t)*qY.sd)
+sc0y)/(max_y+sc0y); //qs'a3   proportion um am
                                     float pr_yam  =
(((qY.am+qY.a3*qY.sd)+(qY.sga3*ci_tp)*qY.sd) +sc0y)/(max_y+sc0y); //qs'a3   proportion um a3
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qs'a3 + y
                                              pr_yam-=dy_sy;

                                               ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                               ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);

                                              //pr_yam  = ((qY.am-
(qY.sga3*ci_t)*qY.sd)+sc0y)/(max_y+sc0y); //qs'a3   proportion um am
                                              pr_yam  = (((qY.am+qY.a3*qY.sd)-
(qY.sga3*ci_tp)*qY.sd)+sc0y)/(max_y+sc0y); //qs'a3   proportion um a3

                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qs'a3 - y
                                              pr_yam-=dy_sy;

                                               ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                               ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(sa3g_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  ôa3 wert y
                              }

                              if(sw_inv==1)//F-1(p)
                              {
                                     //float pr_xam  = ((qY.am+(qY.sga3*ci_t)*qY.sd)
+sc0x)/(max_x+sc0x); //qs'a3   proportion um am
```

```
                                     float pr_xam  = 
(((qY.am+qY.a3*qY.sd)+(qY.sga3*ci_tp)*qY.sd) +sc0x)/(max_x+sc0x); //qs'a3   proportion um a3
                                              pr_xam= pr_xam * dx_s ; //dialog 
koordinatenpunkt qs'a3+ x
                                              pr_xam+=dx_sy;

                                               ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+ 
posY);
                                               ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+ 
posY);
                                     if(sa3g_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-
19+ posY,  cr_);//  ôa3 wert x

                                              //pr_xam  = ((qY.am-(qY.sga3*ci_t)*qY.sd)
+sc0x)/(max_x+sc0x); //qs'a3   proportion um am
                                              pr_xam  = (((qY.am+qY.a3*qY.sd)-
(qY.sga3*ci_tp)*qY.sd) +sc0x)/(max_x+sc0x); //qs'a3   proportion um a3
                                              pr_xam= pr_xam * dx_s ; //dialog 
koordinatenpunkt qs'a3 - x
                                              pr_xam+=dx_sy;

                                     ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+ posY);
                                     ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+ posY);
                              }
                      }

                      if(sw_mod_==3)if(sga4_) // F(p) sga4 linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_sga4); //textfarbe
                                                                      CString cr_="t·ôa4";
                              ooo.SelectObject(&op10);         if(sa4g_q==1)cr_+="= ";

if(sa4g_q==1)cr_+=ftoc(qY.sga4*ci_tp,3);
                                                              if(sa4g_q==1)cr_+="z";
                              if(sw_inv==0) //F(p)
                              {
                                     //float pr_yam  = ((qY.am+(qY.sga4*ci_t)*qY.sd)
+sc0y)/(max_y+sc0y); //qs'a4   proportion um am
                                     float pr_yam  = 
(((qY.am+qY.a4*qY.sd)+(qY.sga4*ci_tp)*qY.sd) +sc0y)/(max_y+sc0y); //qs'a4   proportion um a4
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog 
koordinatenpunkt qs'a4 + y
                                              pr_yam-=dy_sy;

                                               ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+ 
posY);
                                               ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+ 
posY);

                                              //pr_yam  = ((qY.am-(qY.sga4*ci_t)*qY.sd)
+sc0y)/(max_y+sc0y); //qs'a4   proportion um am
                                              pr_yam  = (((qY.am+qY.a4*qY.sd)-
(qY.sga4*ci_tp)*qY.sd) +sc0y)/(max_y+sc0y); //qs'a4   proportion um a4
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog 
koordinatenpunkt qs'a4 - y
                                              pr_yam-=dy_sy;

                                               ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+ 
posY);
                                               ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+ 
posY);
                                     if(sa4g_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+ 
posY,  cr_);//  ôa4 wert y
                              }

                              if(sw_inv==1)//F-1(p)
                              {
                                     //float pr_xam  = ((qY.am+(qY.sga4*ci_t)*qY.sd)
+sc0x)/(max_x+sc0x); //qs'a4   proportion um am
                                     float pr_xam  = 
(((qY.am+qY.a4*qY.sd)+(qY.sga4*ci_tp)*qY.sd) +sc0x)/(max_x+sc0x); //qs'a4   proportion um a4
                                              pr_xam= pr_xam * dx_s ; //dialog 
koordinatenpunkt qs'a4+ x
                                              pr_xam+=dx_sy;

                                               ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+ 
posY);
```

```
                                               ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(sa4g_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-
19+ posY,  cr_);//  ôa4 wert x

                                              //pr_xam  = ((qY.am-(qY.sga4*ci_t)*qY.sd)
+sc0x)/(max_x+sc0x); //qs'a4   proportion um am
                                              pr_xam  = (((qY.am+qY.a4*qY.sd)-
(qY.sga4*ci_tp)*qY.sd) +sc0x)/(max_x+sc0x); //qs'a4   proportion um a4
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qs'a4 - x
                                              pr_xam+=dx_sy;

                                     ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+ posY);
                                     ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+ posY);
                              }
                      }

                   if(sw_mod_==3)if(ag3_) // F(p) ag3 linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_ag3); //textfarbe
                                                                 CString cr_="â3";
                              ooo.SelectObject(&op6);           if(ag3_q==1)cr_+="= ";
                                                         if(ag3_q==1)cr_+=ftoc(qY.ag3,3);
                                                         if(ag3_q==1)cr_+="z";
                              if(sw_inv==0) //F(p)
                              {
                                     float pr_yam  = ((qY.am+qY.ag3*qY.sd)
+sc0y)/(max_y+sc0y); //qa3'   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qa3' y
                                              pr_yam-=dy_sy;

                                              ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                              ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(ag3_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  â3 wert y
                              }

                              if(sw_inv==1)//F-1(p)
                              {
                                     float pr_xam  = ((qY.am+qY.ag3*qY.sd)
+sc0x)/(max_x+sc0x); //qa3'   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qa3' x
                                              pr_xam+=dx_sy;

                                              ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                              ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(ag3_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-19+
posY,  cr_);//  â3 wert x
                              }
                      }

                      if(sw_mod_==3)if(ag4_) // F(p) ag4 linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_ag4); //textfarbe
                                                                 CString cr_="â4";
                              ooo.SelectObject(&op9);           if(ag4_q==1)cr_+="= ";
                                                         if(ag4_q==1)cr_+=ftoc(qY.ag4,3);
                                                         if(ag4_q==1)cr_+="z";
                              if(sw_inv==0) //F(p)
                              {
                                     float pr_yam  = ((qY.am+qY.ag4*qY.sd)
+sc0y)/(max_y+sc0y); //qa4'   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qa4'+ y
                                              pr_yam-=dy_sy;

                                              ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
```

```
                                              ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);

                                              pr_yam  = ((qY.am-qY.ag4*qY.sd)
+sc0y)/(max_y+sc0y); //qa4'   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qa4'- y
                                              pr_yam-=dy_sy;

                                              ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                              ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(ag4_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  â4 wert y
                              }

                              if(sw_inv==1)//F-1(p)
                              {
                                     float pr_xam  = ((qY.am+qY.ag4*qY.sd)
+sc0x)/(max_x+sc0x); //qa4'   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qa4'+ x
                                              pr_xam+=dx_sy;

                                              ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                              ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);

                                              pr_xam  = ((qY.am-qY.ag4*qY.sd)
+sc0x)/(max_x+sc0x); //qa4'   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qa4'- x
                                              pr_xam+=dx_sy;

                                              ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                              ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(ag4_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-19+
posY,  cr_);//  â4 wert x
                              }
                       }


       ////////////////////////////////////////////////////////////////////////////////
/////////////////////////
                       // THETA Fe Achsen zeichnen
                       //

                       if(sw_mod_==4)if(x0_) // F(e) µ0 linie
                       {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_am); //textfarbe
                                                        CString cr_="µ0";
                              ooo.SelectObject(&op13);       if(x0_q==1)cr_+="=";
                                                        if(x0_q==1)cr_+=ftoc(qX.am,3);

                              if(sw_inv==1) //F-1(e)
                              {
                                     float pr_yam  = (qX.am +sc0y)/(max_y+sc0y); //qam
proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qam y
                                              pr_yam-=dy_sy;

                                             ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                             ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(x0_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  µ0 wert y
                              }

                              if(sw_inv==0)//F(e)
                              {
                                     float pr_xam  = (qX.am +sc0x)/(max_x+sc0x); //qam
proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qam x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(x0_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-10+
posY,  cr_);//  µ0 wert x
                              }
                      }

                      if(sw_mod_==4)if(x1_) // F(e) µ1 linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_am); //textfarbe
                                                           CString cr_="µ1";
                              ooo.SelectObject(&op14);       if(x1_q==1)cr_+="=";
                                                           if(x1_q==1)cr_+=ftoc(qY.am,3);

                              if(sw_inv==1) //F-1(e)
                              {
                                     float pr_yam  = (qY.am +sc0y)/(max_y+sc0y); //qam
proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qam y
                                              pr_yam-=dy_sy;

                                             ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                             ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(x1_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  µ1 wert y
                              }

                              if(sw_inv==0)//F(e)
                              {
                                     float pr_xam  = (qY.am +sc0x)/(max_x+sc0x); //qam
proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qam x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(x1_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-10+
posY,  cr_);//  µ1 wert x
                              }
                      }

                      if(sw_mod_==4)if(s0_) // F(e) s0 linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_s0); //textfarbe
                                                                CString cr_="□·s0";
                              ooo.SelectObject(&op15);         if(s0_q==1)cr_+="= ";
                                                         if(s0_q==1)cr_+=ftoc(qX.sd*ci_ze,3);

                              if(sw_inv==1) //F-1(e)
                              {
                                     float pr_yam  = (qX.am+(qX.sd)*ci_ze +sc0y)/(max_y+sc0y);
//qs0   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qs0+ y
                                              pr_yam-=dy_sy;

                                              ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                              ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(s0_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  s0 wert y
```

```
                                              pr_yam  = (qX.am-(qX.sd)*ci_ze
+sc0y)/(max_y+sc0y); //qs0   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qs0- y
                                              pr_yam-=dy_sy;

                                             ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                             ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(s0_q) ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  s0 wert y
                              }

                              if(sw_inv==0)//F(e)
                              {
                                     float pr_xam  = (qX.am+(qX.sd)*ci_ze +sc0x)/(max_x+sc0x);
//qs0   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qs0+ x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(s0_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-10+
posY,  cr_);//  s0 wert x

                                              pr_xam  = (qX.am-(qX.sd)*ci_ze
+sc0x)/(max_x+sc0x); //qs0   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qs0 - x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                     if(s0_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-10+
posY,  cr_);//  s0 wert x

                              }
                      }

                      if(sw_mod_==4)if(s1_) // F(e) s1 linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_s1); //textfarbe
                                                                  CString cr_="□·s1";
                              ooo.SelectObject(&op16);        if(s1_q==1)cr_+="= ";
                                                         if(s1_q==1)cr_+=ftoc(qY.sd*ci_ze,3);

                              if(sw_inv==1) //F-1(e)
                              {
                                     float pr_yam  = (qY.am+(qY.sd)*ci_ze +sc0y)/(max_y+sc0y);
//qs1   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qs1+ y
                                              pr_yam-=dy_sy;

                                             ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                             ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                     if(s0_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  s1 wert y

                                              pr_yam  = (qY.am-(qY.sd)*ci_ze
+sc0y)/(max_y+sc0y); //qs1   proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qs1- y
                                              pr_yam-=dy_sy;

                                             ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
```

```
                                             ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                    if(s0_q) ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  s0 wert y
                             }

                             if(sw_inv==0)//F(e)
                             {
                                    float pr_xam  = (qY.am+(qY.sd)*ci_ze +sc0x)/(max_x+sc0x);
//qs1   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qs1+ x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                    if(s0_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-10+
posY,  cr_);//  s1 wert x

                                              pr_xam  = (qY.am-(qY.sd)*ci_ze
+sc0x)/(max_x+sc0x); //qs1   proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qs1 - x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                    if(s0_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-10+
posY,  cr_);//  s1 wert x
                             }
                      }


                      if(sw_mod_==4)if(xc_) // F(e) xcrit linie
                      {
                             ooo.SelectObject(&of1); //font y-achse
                             ooo.SetTextColor(fb_am); //textfarbe
                                                          CString cr_="xcrit";
                             ooo.SelectObject(&op12);       if(x_q==1)cr_+="=";
                                                          if(x_q==1)cr_+=ftoc(qE.sw,3);

                             if(sw_inv==1) //F-1(e)
                             {
                                    float pr_yam  = (qE.sw +sc0y)/(max_y+sc0y); //qsw
proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qam y
                                              pr_yam-=dy_sy;

                                            ooo.MoveTo (x_m_min*frmX+ posX, pr_yam*frmY+
posY);
                                            ooo.LineTo (x_m_max*frmX+ posX, pr_yam*frmY+
posY);
                                    if(x_q)ooo.TextOut(x_m_max*frmX+3+ posX,pr_yam*frmY-5+
posY,  cr_);//  xcrit wert y
                             }

                             if(sw_inv==0)//F(e)
                             {
                                    float pr_xam  = (qE.sw +sc0x)/(max_x+sc0x); //qsw
proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qam x
                                              pr_xam+=dx_sy;

                                            ooo.MoveTo (pr_xam*frmX+ posX,y_m_min*frmY+
posY);
                                            ooo.LineTo (pr_xam*frmX+ posX,y_m_max*frmY+
posY);
                                    if(x_q)ooo.TextOut(pr_xam*frmX-1+ posX,y_m_min*frmY-20+
posY,  cr_);//  xcrit wert x
                             }
                      }
                      if(sw_mod_==4)if(e_) // F(e) e linie
                      {
                              ooo.SelectObject(&of1); //font y-achse
                              ooo.SetTextColor(fb_am); //textfarbe
                                                           CString cr_="e";
                              ooo.SelectObject(&op11);      if(e_q==1)cr_+="=";
                                                           if(e_q==1)cr_+=ftoc(qE.e,3);

                              if(sw_inv==1) //F-1(e)
                              {
                                     float pr_yam  = (qX.am +sc0y)/(max_y+sc0y); //qam
proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qam y
                                              pr_yam-=dy_sy;

                                     float pr_yam1  = (qY.am +sc0y)/(max_y+sc0y); //qam
proportion
                                              pr_yam1= dlg.y - ( pr_yam1 * dy_s ); //dialog
koordinatenpunkt qam y
                                              pr_yam1-=dy_sy;

                                     float pr_xam  = (0.5 +sc0x)/(max_x+sc0x); //0.5
proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qam x
                                              pr_xam+=dx_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX, pr_yam*frmY+
posY);
                                             ooo.LineTo (pr_xam*frmX+ posX, pr_yam1*frmY+
posY);
                                     if(e_q)ooo.TextOut(pr_xam*frmX+
posX,((pr_yam+pr_yam1)/2)*frmY-15+ posY,  cr_);//  e wert y
                              }

                              if(sw_inv==0)//F(e)
                              {
                                     float pr_xam  = (qX.am +sc0x)/(max_x+sc0x); //qam
proportion
                                              pr_xam= pr_xam * dx_s ; //dialog
koordinatenpunkt qam x
                                              pr_xam+=dx_sy;

                                     float pr_xam1  = (qY.am +sc0x)/(max_x+sc0x); //qam
proportion
                                              pr_xam1= pr_xam1 * dx_s ; //dialog
koordinatenpunkt qam x
                                              pr_xam1+=dx_sy;

                                     float pr_yam  = (0.5 +sc0y)/(max_y+sc0y); //0.5
proportion
                                              pr_yam= dlg.y - ( pr_yam * dy_s ); //dialog
koordinatenpunkt qam y
                                              pr_yam-=dy_sy;

                                             ooo.MoveTo (pr_xam*frmX+ posX,pr_yam*frmY+ posY);
                                             ooo.LineTo (pr_xam1*frmX+ posX,pr_yam*frmY+
posY);
                                     if(e_q)ooo.TextOut(((pr_xam+pr_xam1)/2)*frmX+
posX,pr_yam*frmY-15+ posY,  cr_);//  e wert x
                              }
                      }



       ////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////
                      // ACHSEN Skalierung, Skalenwerte und Grid zeichnen
                      //

                      ooo.SelectObject(&o3s);

                      if(sw_xS==1)// x achsen skalen markierung
                      {

float dx= ((max_x-min_x)      /tlg_x);
float dx1=dx;



float x_m_1=((x_m_max-x_m_min)/tlg_x);
                          for(float x_m= x_m_min ;x_m<=x_m_max;x_m+=x_m_1 )
                          {
                                 ooo.MoveTo (x_m*frmX+ posX,dapty*frmY-2+posSCx+
posY);//skalenmarkierung x
                                 ooo.LineTo (x_m*frmX+ posX,dapty*frmY+2+posSCx+ posY);

                                 if(sw_Grdx)//grid x
                                 {    ooo.SelectObject(&ogx);
                                             ooo.LineTo (x_m*frmX+ posX,y_m_min*frmY+ posY);
                                 }

                                 if(sw_xSw==1)// skalenwerte x
                                 {
                                         ooo.SelectObject(&of2);
                                         ooo.SetTextColor(Ax_fb);
                                         ooo.SetBkMode(TRANSPARENT);

                                         if(x_m== x_m_min) ooo.TextOut(x_m*frmX +posAXx+
posX,dapty*frmY +posAXy+ posY, ftoc(min_x,ds_xSw) );
                                         if(x_m!= x_m_min )
                                         {
                                                 ooo.TextOut(x_m*frmX +posAXx+ posX,dapty*frmY
+posAXy+ posY,  ftoc((min_x)+dx1,ds_xSw));

                                               dx1+=dx;
                                         }
                                 }
                          }
                  }

                  ooo.SelectObject(&o4s);

                  if(sw_yS==1)// y achsen skalen markierung
                  {

float dy= ((max_y-min_y)      /tlg_y);

float dy1=dy;



float y_m_1=((y_m_max-y_m_min)/tlg_y);
                          for(float y_m= y_m_min ;y_m<=y_m_max;y_m+=y_m_1 )
                          {
                                 ooo.MoveTo (daptx*frmX-2+posSCy+ posX, y_m*frmY+
posY);//skalenmarkierung y
                                 ooo.LineTo (daptx*frmX+2+posSCy+ posX, y_m*frmY+ posY);

                                 if(sw_Grdy)//grid y
                                 {
                                         ooo.SelectObject(&ogy);
                                         ooo.LineTo (x_m_max*frmX+ posX, y_m*frmY+ posY);
                                 }

                                 if(sw_ySw==1)// skalenwerte y
                                 {
                                         ooo.SelectObject(&of1);
                                         ooo.SetTextColor(Ay_fb);
                                         ooo.SetBkMode(TRANSPARENT);

                                         if(y_m== y_m_min) ooo.TextOut(daptx*frmX +posAYx+
posX, y_m*frmY +posAYy+ posY,  ftoc((min_y+max_y)-min_y,ds_ySw));
                                         if(y_m!= y_m_min )
                                         {
                                                 ooo.TextOut(daptx*frmX +posAYx+ posX, y_m*frmY
+posAYy+ posY,  ftoc((min_y+max_y)-((min_y)+dy1),ds_ySw));

                                                                             dy1+=dy;
                                         }
                                 }
                          }
                  }
```

```
          }//sw_drw / rerendern //



/////////////////////////////////////////////////////////////////////////////////////////////
///////////////////
          // VEKTOREN und Vektor-Koordinaten zeichnen
          //

               //Vektorkoordinatenpunkte
                            float dvptx;
                            float dvpty;
               if(sw_v0==0)if(!sw_mkoord_V)//
               {
                                      dvptx = egx *(mv1x/(e_x/sc)); //dialog-vektor-
koordinatenpunkt x
                                  dvpty = egy *(mv1y/(e_y/sc)); //dialog-vektor-
koordinatenpunkt y
               }
               if(sw_v0==1)if(!sw_mkoord_V)//bei r(x,y) (0,0)
               {
                                      dvptx = (        ((((0)) +sc0x)/(max_x+sc0x))*
dx_s)+dx_sy; //dialog-vektor-koordinatenpunkt x0
                                      dvpty = ( dlg.y -((((0)) +sc0y)/(max_y+sc0y))* dy_s)-
dy_sy; //dialog-vektor-koordinatenpunkt y0
               }
               if(sw_v0==2)if(!sw_mkoord_V)//bei F(p) (0.5,y/2)
               {
                         if(sw_inv==0)dvptx = (                    ((((0.5))
+sc0x)/(max_x+sc0x))* dx_s)+dx_sy; //dialog-vektor-koordinatenpunkt x0.5
                         if(sw_inv==0)dvpty = ( dlg.y -(((((max_y+min_y)/2))
+sc0y)/(max_y+sc0y))* dy_s)-dy_sy; //dialog-vektor-koordinatenpunkt y/2
                         if(sw_inv==1)dvptx = (        ((((max_x+min_x)/2))
+sc0x)/(max_x+sc0x))* dx_s)+dx_sy; //dialog-vektor-koordinatenpunkt x/2
                         if(sw_inv==1)dvpty = (             dlg.y -((((0.5))
+sc0y)/(max_y+sc0y))* dy_s)-dy_sy; //dialog-vektor-koordinatenpunkt y0.5
               }
               if(sw_v0==3)if(!sw_mkoord_V)//bei f(x) (xmax,ymax)
               {
                                      dvptx = (         (((max_x))  +sc0x)/(max_x+sc0x))*
dx_s)+dx_sy; //dialog-vektor-koordinatenpunkt xmax
                                      dvpty = ( dlg.y -(((((max_y))) +sc0y)/(max_y+sc0y))*
dy_s)-dy_sy; //dialog-vektor-koordinatenpunkt ymax
               }
               if(sw_v0==4)if(!sw_mkoord_V)//bei F(e) (xcrit,pmax)
               {
                        if(sw_inv==0)dvptx = (         ((((qE.sw)) +sc0x)/(max_x+sc0x))*
dx_s)+dx_sy; //dialog-vektor-koordinatenpunkt xcrit
                        if(sw_inv==0)dvpty = ( dlg.y -(((((max_y)))+sc0y)/(max_y+sc0y))* dy_s)-
dy_sy; //dialog-vektor-koordinatenpunkt pmax
                        if(sw_inv==1)dvptx = (        (((((max_x))) +sc0x)/(max_x+sc0x))*
dx_s)+dx_sy; //dialog-vektor-koordinatenpunkt pmax
                        if(sw_inv==1)dvpty = (  dlg.y -((((qE.sw)) +sc0y)/(max_y+sc0y))* dy_s)-
dy_sy; //dialog-vektor-koordinatenpunkt xcrit
               }

               if(sw_mkoord_V)//bei manueller vektor koordinatenbestimmung
               {
                                      dvptx = (         (((mVx))  +sc0x)/(max_x+sc0x))*
dx_s)+dx_sy; //dialog-vektor-koordinatenpunkt x manuell
                                      dvpty = ( dlg.y -(((((mVy))) +sc0y)/(max_y+sc0y))* dy_s)-
dy_sy; //dialog-vektor-koordinatenpunkt     y manuell
               }

               ooo.SelectObject(&o1);
               if(sw_xV==1)
               ooo.MoveTo(        dvptx*frmX+ posX, 0);     // vektor x
               if(sw_xV==1)
               ooo.LineTo(        dvptx*frmX+ posX, dlg.y);

               ooo.SelectObject(&o2);
               if(sw_yV==1)
               ooo.MoveTo( 0,     dvpty*frmY+ posY);        // vektor y
               if(sw_yV==1)
               ooo.LineTo( dlg.x, dvpty*frmY+ posY);

               //Vektorkoordinatenwerte
               double  xy_x = (((( egx*(    mv1x/(e_x/sc)))-dx_sy)/dx_s)*(max_x+sc0x))-sc0x;
               //double  xy_y=  ((((dlg.y-(dlg.y/15.0))-
(egy*(mv1y/(e_y/sc)))+dy_sy)/dy_s)*(max_y+sc0y))-sc0y; //alternativ
               double  xy_y = (((( egy*((1-mv1y)/(e_y/sc)))-dy_sy)/dy_s)*(max_y+sc0y))-sc0y;

               if(sw_v0==1){             xy_x=0;                xy_y=0;}               //bei
r(x,y) (0,0)
               if(sw_v0==2)if(sw_inv==0){xy_x=0.5;              xy_y=(max_y+min_y)/2;} //bei
F(p) (0.5,y/2)
               if(sw_v0==2)if(sw_inv==1){xy_x=(max_x+min_x)/2; xy_y=0.5;}             //bei
F(q) (x/2,0.5)
               if(sw_v0==3){             xy_x=max_x;            xy_y=max_y;}           //bei
f(x) (xmax,ymax)
               if(sw_v0==4)if(sw_inv==0){xy_x=qE.sw;            xy_y=max_y;}           //bei
F(e) (xcrit,pmax)
               if(sw_v0==4)if(sw_inv==1){xy_x=max_x;            xy_y=qE.sw;}           //bei
F(e) (pmax,xcrit)

                                         xy_x+=corx;            xy_y+=cory;
               //                       Vx_m=xy_x;            Vy_m=xy_y;
//global für koordinatenübergabe an koordinateneinstellungsdialog
               //if(!sw_mkoord_V){mVx=Vx_m;             mVy=Vy_m;     }                //nicht
bei manueller koordinateneinstellung
               if(!sw_mkoord_V){     mVx=xy_x;             mVy=xy_y;     } //Vx_m, Vy_m
überflüssig...

                                                                CString c="(";
               if(sw_inv==0)if(sw_xK==1)  { if(sw_mod_==1)c+="x:"; if(sw_mod_==2)c+="zx:";
if(sw_mod_==5)c+="zx:";if(sw_mod_==3)c+="p:";if(sw_mod_==4)c+="q:";}
               if(sw_inv==1)if(sw_xK==1)  { if(sw_mod_==1)c+="y:"; if(sw_mod_==2)c+="zx:";
if(sw_mod_==5)c+="zy:";if(sw_mod_==3)if(sw_pq==0)c+="x:";if(sw_mod_==3)if(sw_pq==1)c+="y:";if(
sw_mod_==4)c+="p:";}//f-1(x)
               if(sw_xK==1)if(!sw_mkoord_V)               c+=ftoc(xy_x,ds_xk);
               if(sw_xK==1)if( sw_mkoord_V)               c+=ftoc(mVx,ds_xk);
               if(sw_xK==1&&sw_yK==1)                     c+="; ";
               if(sw_inv==0)if(sw_yK==1)  { if(sw_mod_==1)c+="y:"; if(sw_mod_==2)c+="zy:";
if(sw_mod_==5)c+="zy:";if(sw_mod_==3)if(sw_pq==0)c+="x:";if(sw_mod_==3)if(sw_pq==1)c+="y:";if(
sw_mod_==4)c+="p:";}
               if(sw_inv==1)if(sw_yK==1)  { if(sw_mod_==1)c+="x:"; if(sw_mod_==2)c+="zy:";
if(sw_mod_==5)c+="zx:";if(sw_mod_==3)c+="p:";if(sw_mod_==4)c+="q:";} //f-1(x)
               if(sw_yK==1)if(!sw_mkoord_V)               c+=ftoc(xy_y,ds_yk);
               if(sw_yK==1)if( sw_mkoord_V)               c+=ftoc(mVy,ds_yk);
                                                                     c+=")";
               ooo.SelectObject(&ofv);
               ooo.SetTextColor(V_fb);
               ooo.SetBkMode(2);                                //OPAQUE koordinatenwerte vor
die funktion
               if(sw_xK||sw_yK)
               ooo.TextOut(dvptx*frmX+ posVx+ posX, dvpty*frmY +posVy+ posY,    c  );




        ////////////////////////////////////////////////////////////////////////////////////////
////////////////

               ooo.SelectObject(&ofx);
               ooo.SetTextColor(fn_x_fb);
               ooo.TextOut(5, dlg.y-12,"SCHRAUSSER-MAT");
               ooo.TextOut(dlg.x-90, dlg.y-12,"(C) SCHRAUSSER");
                                c="[";
                                c+=filename_tmp;
                                              c+="]";
               ooo.TextOut(5, 0,c);
                                  char datum[9];
                          _strdate( datum );
                                      char zeit[9];
                                 _strtime( zeit );
                             strcat(datum, " ");
                             strcat(datum, zeit);
               ooo.TextOut(dlg.x-90, 0,datum);
               ooo.CloseEnhanced();

               sw_emf=0;
}
```

References

Schrausser, D. G. (2022). *SCHRAUSSER-MAT: Funktionsindex.* DOI:10.13140/RG.2.2.28314.52164

Schrausser, D. G. (2023). *Schrausser/GRP2: GRP2 (v1.0.0)*. Zenodo. DOI:10.5281/zenodo.7647467