

There are many books on the market that discuss the Mandelbrot set, Julia sets, and fractals in general. Below is a list of several I own that you might find interesting.

- Barnsley, Michael; Fractals Everywhere; Academic Press 1988.
 Barnsley, Michael F. and Hurd, Lyman P.; Fractal Image Compression; AK Peters 1993.
 Cvitanovic, Predrag; Universality in Chaos; Adam Hilger Ltd. 1984.
 Feder, Jens; Fractals; Plenum Press 1988.
 Gleick, James; Chaos: Making a New Science; Viking 1987.
 Mandelbrot, Benoit B.; The Fractal Geometry of Nature; W.H. Freeman & Co. 1983.
 Oliver, Dick; FractalVision: Put Fractals to Work for You; SAMS Publishing 1992.
 Peitgen, Heinz-Otto, Jurgens, Hartmut, and Saupe, Dietmar; Fractals for the Classroom: Part I Introduction to Fractals and Chaos; Springer-Verlag 1992.
 Peitgen, H.-O. and Richter, P.H.; The Beauty of Fractals; Springer-Verlag 1986.
 Peitgen, H.-O. and Saupe, D.; The Science of Fractal Images; Springer-Verlag 1988.
 Sparrow, Colin; The Lorenz Equations: Bifurcations, Chaos, and Strange Attractors; Springer-Verlag 1982.
 Stevens, Roger T.; Fractal Programming in C; M&T Publishing, Inc. 1989.
 Thompson, J.M.T. and Stewart, H.B.; Nonlinear Dynamics and Chaos; John Wiley and Sons 1986.

Listings

Listing 1

Listing1. CanDoMJDeck

```
*****
*Deck`CanDoMJ`
*Time15:36:09
*Date03/05/94
*****

*****
* Card(s) in deck.
* Card`MJPlotPic`
* Card`MJSettings`
*****
* 2 Card(s), 2 were printed.
*****

*****
* Natural order of Cards
* Card`MJSettings`
* Card`MJPlotPic`
*****

*****
* There are no Global routines in this deck.
*****

*****
*Card`MJPlotPic`
AfterAttachment ; used to be AfterStartup
Nop ;Drawpicture using CanDo code
If Arg1=`GO`
  SetDrawMode JAM2
  Let SR=(HR-LR)*Resolution/(WindowWidth-1)
  Let SI=(HI-LI)*Resolution/(WindowHeight-1)
  Let HC=WindowColors-1
  Let LC=0
  Nop ;Begin iterations
  Let X=0
  Let Y=0
  Let R=LR
  Let I=HI
  Nop ;Plot Mandelbrot set
```

```
IfPlotType=`M`
While Y<WindowHeight
  While X<WindowWidth
    Let LoopR=0
    Let LoopI=0
    Let Count=0
    Loop
    Let Temp=LoopR*LoopR-LoopI*LoopI+R
    Let LoopI=2*LoopR*LoopI+I
    Let LoopR=Temp
    If (LoopR*LoopR+LoopI*LoopI)>4.0
      ExitLoop
    EndIf
    Let Count=Count+1
    Until Count>=MaxDwell
    SetPen Count
    If Resolution=1
      DrawPixel X,Y
    Else
      AreaRectangle X,Y,Resolution,Resolution
    EndIf
    Let X=X+Resolution
    Let R=R+SR
  EndLoop
  Let X=0
  Let R=LR
  Let Y=Y+Resolution
  Let I=I-SI
  EndLoop
Nop ;Plot Julia set
ElseIf PlotType=`J`
While Y<WindowHeight
  While X<WindowWidth
    Let LoopR=R
    Let LoopI=I
    Let Count=0
    Loop
    If (LoopR*LoopR+LoopI*LoopI)>4.0
      ExitLoop
    EndIf
    Let Count=Count+1
    Let Temp=LoopR*LoopR-LoopI*LoopI+JR
    Let LoopI=2*LoopR*LoopI+JI
    Let LoopR=Temp
    Until Count>=MaxDwell
    SetPen Count
    If Resolution=1
      DrawPixel X,Y
    Else
      AreaRectangle X,Y,Resolution,Resolution
    EndIf
    Let X=X+Resolution
    Let R=R+SR
  EndLoop
  Let X=0
  Let R=LR
  Let Y=Y+Resolution
  Let I=I-SI
  EndLoop
EndIf
Nop ;Drawpicture using C code
ElseIf Arg1=`GCC`
Let Command=`RCF:C Progs/CanDoMJC/CanDoMJC`
Let Command=Command|WindowAddress|`
Let Command=Command|MaxDwell|`
Let Command=Command|Resolution|`
Let Command=Command|LR|`|HR|`|LI|`|HI|`
Let Command=Command|PlotType|`
Let Command=Command|JR|`|JI
Dos Command
ElseIf Arg1=`SEE`
LoadPicture`ram:CanDoMPic`,`MPic`
ShowPicture`MPic`
EndIf
EndScript
Window`UserWindow`
Definition
Origin 0,0
Size 320,200
Title ``
NumberOfColors 32,69632
WindowColors 0,1,0 ; Detail, Block, Background
WindowObjects NONE
WindowFlags ACTIVATE BORDERLESS SEPARATE SCREEN TO FRONT
EndScript
OnCloseButton
Quit
EndScript
EndObject
AreaButton`NewRange`
Definition
Origin 0,0
Size 320,200
Border NONE,2,1 ; BorderStyle, MainPen, ExtraPen
Highlight NONE
```

```

ButtonFlags NONE
EndScript
OnClick
  Let BeginX=MouseX
  Let BeginY=MouseY
  Let CurrW=0
  Let CurrH=0
  SetDrawMode COMPLEMENT
  DrawRectangle BeginX, BeginY, CurrW, CurrH
EndScript
OnDrag
  If MouseX < (CurrW+BeginX) or MouseY < (CurrH+BeginY)
  DrawRectangle BeginX, BeginY, CurrW, CurrH
    Let CurrW=MouseX-BeginX
    Let CurrH=MouseY-BeginY
  DrawRectangle BeginX, BeginY, CurrW, CurrH
  EndIf
EndScript
OnRelease
  DrawRectangle BeginX, BeginY, CurrW, CurrH
  Let Temp=LR+(HR-IR)*BeginX/(WindowWidth-1)
  Let HR=LR+(HR-IR)*(BeginX+CurrW)/(WindowWidth-1)
  Let LR=Temp
  Let Temp=HI-(HI-LI)*(BeginY+CurrH)/(WindowHeight-1)
  Let HI=HI-(HI-LI)*BeginY/(WindowHeight-1)
  Let LI=Temp
  ClipPicture "MOPic"
  SavePicture "MPic", "ram:CanDoMOPic"
  GotoCard "MSettings", "Reset"
EndScript
EndObject
TextMenu "SavePic"
  Definition
  AttachToMENU, "Options"
  Font "topaz", 8; FontName, PointSize
  PrintStyle PLAIN, 2, 3; Style, Pen1, Pen2
  TextColors 0, 1, NORMAL; PenA, PenB, DrawMode
  Text "Save Picture"
  MenuFlags NONE
  Highlight COMPLEMENT
  ShortCutKey "S"
EndScript
Occurred
  Let FNNew=AskForFileName(FN, "SavePicture")
  If FNNew < ""
    Let FN=FNNew
    ClipPicture "MPic"
    SavePicture "MPic", FN
  EndIf
EndScript
EndObject
TextMenu "SetColors"
  Definition
  AttachToMENU, "Options"
  Font "topaz", 8; FontName, PointSize
  PrintStyle PLAIN, 2, 3; Style, Pen1, Pen2
  TextColors 0, 1, NORMAL; PenA, PenB, DrawMode
  Text "Set Colors"
  MenuFlags NONE
  Highlight COMPLEMENT
  ShortCutKey "C"
EndScript
Occurred
  LoadSubDeck "CanDo:Decks/ColorChange", "CC"
  OpenRequester "CC", "ChangePalette"
EndScript
EndObject
TextMenu "ToMSettings"
  Definition
  AttachToMENU, "Options"
  Font "topaz", 8; FontName, PointSize
  PrintStyle PLAIN, 2, 3; Style, Pen1, Pen2
  TextColors 0, 1, NORMAL; PenA, PenB, DrawMode
  Text "To Settings"
  MenuFlags NONE
  Highlight COMPLEMENT
  ShortCutKey "T"
EndScript
Occurred
  ClipPicture "MOPic"
  SavePicture "MPic", "ram:CanDoMOPic"
  GotoCard "MSettings", "Reset"
EndScript
EndObject
*EndOfCard "MPlotPic"
*****
*****
*Card "MSettings"
BeforeAttachment; used to be OnStartup
If Invocation=0
  Let MaxDwell=15
  Let Resolution=1
  Let PlotType="M"
  EndIf

```

```

  Let Invocation=1
EndScript
AfterAttachment; used to be AfterStartup
  Nop; Remove checkmark from all menu subitems
  SetObjectState "MD15", OFF
  SetObjectState "MD31", OFF
  SetObjectState "MD63", OFF
  SetObjectState "MD127", OFF
  SetObjectState "RES1", OFF
  SetObjectState "RES2", OFF
  SetObjectState "RES4", OFF
  SetObjectState "RES8", OFF
  SetObjectState "Mandelbrot", OFF
  SetObjectState "Julia", OFF
  Nop; Put checkmark on appropriate menu subitems
  SetObjectState "MD" || MaxDwell, ON
  SetObjectState "RES" || Resolution, ON
  If PlotType="M"
    SetObjectState "Mandelbrot", ON
  Else
    SetObjectState "Julia", ON
  EndIf
  Nop; Set text entries if they changed
  If Arg1="Reset"
    SetText "LowReal", IR
    SetText "HighReal", HR
    SetText "LowImag", LI
    SetText "HighImag", HI
    SetText "JReal", JR
    SetText "JImag", JI
  EndIf
  SetPrintFont "CGTimes", 20
  SetPrintStyle EMBOSSED, 2, 11
  SetPen 1, 0
  SetDrawMode JAMI
  PrintText "MANDELBROT OR JULIA SET BOUNDARIES", 220, 18
  SetPrintFont "Pearl", 11
  SetPrintStyle BOLD EMBOSSED, 2, 6
  SetPen 1, 0
  SetDrawMode JAMI
  PrintText "  LowReal Value:", 227, 56
  PrintText "  HighReal Value:", 227, 86
  PrintText "  LowImaginary Value:", 227, 116
  PrintText "  HighImaginary Value:", 227, 146
  PrintText "  Julia Real Value:", 12, 168
  PrintText "  Julia Imag Value:", 12, 183
  SetPrintFont "Pearl", 8
  SetPrintStyle EMBOSSED, 2, 3
  SetPen 1, 0
  SetDrawMode JAMI
  PrintText "SeePicture", 73, 149
EndScript
Routine "DrawMOPic"
  Let IR=TextFrom("LowReal")
  Let HR=TextFrom("HighReal")
  Let LI=TextFrom("LowImag")
  Let HI=TextFrom("HighImag")
  Let JR=TextFrom("JReal")
  Let JI=TextFrom("JImag")
  GotoCard "MPlotPic", Arg1
EndScript
Window "UserWindow"
  Definition
  Origin 0, 0
  Size 640, 200
  Title "CanDoM"
  NumberOfColors 16, 102400
  WindowColors 0, 1, 0; Detail, Block, Background
  WindowObjects CLOSEBUTTON
  WindowFlags ACTIVATE SEPARATE SCREEN TO FRONT
  EndScript
  OnCloseButton
  Quit
  EndScript
EndObject
TextButton "G"
  Definition
  Origin 561, 179
  Font "topaz", 8; FontName, PointSize
  PrintStyle PLAIN, 2, 3; Style, Pen1, Pen2
  TextColors 1, 0, NORMAL; PenA, PenB, DrawMode
  Text "GO"
  Border BEVEL, 2, 1; BorderStyle, MainPen, ExtraPen
  Highlight COMPLEMENT
  ButtonFlags NONE
  EndScript
  OnRelease
  Do "DrawMOPic", "GO"
  EndScript
EndObject
TextField "LowReal"
  Definition
  Origin 412, 59
  Size 200, 8
  Justification LEFT

```

```

MaxFieldLength 32
InitialText ``-2.9''
Border DOUBLEBEVEL ,2,1 ; BorderStyle, MainPen, ExtraPen
EndScript
OnRelease
SetObjectState ``HighReal'',ON
EndScript
EndObject
TextField ``HighReal''
Definition
Origin 412,89
Size 200,8
Justification LEFT
MaxFieldLength 32
InitialText ``2.9''
Border DOUBLEBEVEL ,2,1 ; BorderStyle, MainPen, ExtraPen
EndScript
OnRelease
SetObjectState ``LowImag'',ON
EndScript
EndObject
TextField ``LowImag''
Definition
Origin 412,119
Size 200,8
Justification LEFT
MaxFieldLength 32
InitialText ``-2.0''
Border DOUBLEBEVEL ,2,1 ; BorderStyle, MainPen, ExtraPen
EndScript
OnRelease
SetObjectState ``HighImag'',ON
EndScript
EndObject
TextField ``HighImag''
Definition
Origin 412,149
Size 200,8
Justification LEFT
MaxFieldLength 32
InitialText ``2.0''
Border DOUBLEBEVEL ,2,1 ; BorderStyle, MainPen, ExtraPen
EndScript
OnRelease
SetObjectState ``JReal'',ON
EndScript
EndObject
TextField ``JImag''
Definition
Origin 161,186
Size 200,8
Justification LEFT
MaxFieldLength 32
InitialText ``0.0''
Border DOUBLEBEVEL ,2,1 ; BorderStyle, MainPen, ExtraPen
EndScript
OnRelease
SetObjectState ``LowReal'',ON
EndScript
EndObject
TextField ``JReal''
Definition
Origin 161,171
Size 200,8
Justification LEFT
MaxFieldLength 32
InitialText ``0.5''
Border DOUBLEBEVEL ,2,1 ; BorderStyle, MainPen, ExtraPen
EndScript
OnRelease
SetObjectState ``JImag'',ON
EndScript
EndObject
TextButton ``Gc''
Definition
Origin 471,179
Font ``topaz'',8 ; FontName, PointSize
PrintStyle PLAIN ,2,3 ; Style, Pen1, Pen2
TextColors 1,0,NORMAL ; PenA, PenB, DrawMode
Text `` G C ``
Border BEVEL ,2,1 ; BorderStyle, MainPen, ExtraPen
Highlight COMPLEMENT
ButtonFlags NONE
EndScript
OnRelease
Do ``DrawMIPic'',``GC''
EndScript
EndObject
ImageButton ``SeeMIPic''
Definition
Origin 28,19
Image ``CanDo:Brushes/CanDoMButton.br''
Highlight COMPLEMENT
ButtonFlags NONE
EndScript

```

```

OnRelease
Do ``DrawMIPic'',``SEE''
EndScript
EndObject
TextMenu ``MaxDwell''
Definition
AttachToMENU ``PlotInfo''
Font ``topaz'',8 ; FontName, PointSize
PrintStyle PLAIN ,2,3 ; Style, Pen1, Pen2
TextColors 0,1,NORMAL ; PenA, PenB, DrawMode
Text ``MaxDwell''
MenuFlags NONE
Highlight COMPLEMENT
ShortcutKey ``''
EndScript
EndObject
TextMenu ``MD15''
Definition
AttachToOBJECT ``MaxDwell''
Font ``topaz'',8 ; FontName, PointSize
PrintStyle PLAIN ,2,3 ; Style, Pen1, Pen2
TextColors 0,1,NORMAL ; PenA, PenB, DrawMode
Text `` 15 ``
MenuFlags CHECKABLE CHECKIT
Highlight COMPLEMENT
ShortcutKey ``1''
EndScript
Occurred
Let MaxDwell=15
SetObjectState ``MD31'',OFF
SetObjectState ``MD63'',OFF
SetObjectState ``MD127'',OFF
EndScript
EndObject
TextMenu ``MD31''
Definition
AttachToOBJECT ``MaxDwell''
Font ``topaz'',8 ; FontName, PointSize
PrintStyle PLAIN ,2,3 ; Style, Pen1, Pen2
TextColors 0,1,NORMAL ; PenA, PenB, DrawMode
Text `` 31 ``
MenuFlags CHECKABLE
Highlight COMPLEMENT
ShortcutKey ``2''
EndScript
Occurred
Let MaxDwell=31
SetObjectState ``MD15'',OFF
SetObjectState ``MD63'',OFF
SetObjectState ``MD127'',OFF
EndScript
EndObject
TextMenu ``MD63''
Definition
AttachToOBJECT ``MaxDwell''
Font ``topaz'',8 ; FontName, PointSize
PrintStyle PLAIN ,2,3 ; Style, Pen1, Pen2
TextColors 0,1,NORMAL ; PenA, PenB, DrawMode
Text `` 63 ``
MenuFlags CHECKABLE
Highlight COMPLEMENT
ShortcutKey ``3''
EndScript
Occurred
Let MaxDwell=63
SetObjectState ``MD15'',OFF
SetObjectState ``MD31'',OFF
SetObjectState ``MD127'',OFF
EndScript
EndObject
TextMenu ``MD127''
Definition
AttachToOBJECT ``MaxDwell''
Font ``topaz'',8 ; FontName, PointSize
PrintStyle PLAIN ,2,3 ; Style, Pen1, Pen2
TextColors 0,1,NORMAL ; PenA, PenB, DrawMode
Text `` 127 ``
MenuFlags CHECKABLE
Highlight COMPLEMENT
ShortcutKey ``4''
EndScript
Occurred
Let MaxDwell=127
SetObjectState ``MD15'',OFF
SetObjectState ``MD31'',OFF
SetObjectState ``MD63'',OFF
EndScript
EndObject
TextMenu ``Resolution''
Definition
AttachToMENU ``PlotInfo''
Font ``topaz'',8 ; FontName, PointSize
PrintStyle PLAIN ,2,3 ; Style, Pen1, Pen2
TextColors 0,1,NORMAL ; PenA, PenB, DrawMode
Text ``Resolution''

```

```

MenuFlags NONE
Highlight COMPLEMENT
ShortCutKey `` ``
EndScript
EndObject
TextMenu ``RES1``
Definition
AttachToOBJECT, ``Resolution``
Font ``topaz``, 8; FontName, PointSize
PrintStyle PLAIN, 2, 3; Style, Pen1, Pen2
TextColors 0, 1, NORMAL; PenA, PenB, DrawMode
Text `` 1 ``
MenuFlags CHECKABLE CHECKIT
Highlight COMPLEMENT
ShortCutKey ``5``
EndScript
Occurred
Let Resolution=1
SetObjectState ``RES2``, OFF
SetObjectState ``RES4``, OFF
SetObjectState ``RES8``, OFF
EndScript
EndObject
TextMenu ``PlotType``
Definition
AttachToMENU, ``PlotInfo``
Font ``topaz``, 8; FontName, PointSize
PrintStyle PLAIN, 2, 3; Style, Pen1, Pen2
TextColors 0, 1, NORMAL; PenA, PenB, DrawMode
Text ``Plot Type``
MenuFlags NONE
Highlight COMPLEMENT
ShortCutKey `` ``
EndScript
EndObject
TextMenu ``Mandellbrot``
Definition
AttachToOBJECT, ``PlotType``
Font ``topaz``, 8; FontName, PointSize
PrintStyle PLAIN, 2, 3; Style, Pen1, Pen2
TextColors 0, 1, NORMAL; PenA, PenB, DrawMode
Text ``Mandellbrot ``
MenuFlags CHECKABLE CHECKIT
Highlight COMPLEMENT
ShortCutKey ``M``
EndScript
Occurred
Let PlotType= ``M``
SetObjectState ``Julia``, OFF
EndScript
EndObject
TextMenu ``Julia``
Definition
AttachToOBJECT, ``PlotType``
Font ``topaz``, 8; FontName, PointSize
PrintStyle PLAIN, 2, 3; Style, Pen1, Pen2
TextColors 0, 1, NORMAL; PenA, PenB, DrawMode
Text `` Julia ``
MenuFlags CHECKABLE
Highlight COMPLEMENT
ShortCutKey ``J``
EndScript
Occurred
Let PlotType= ``J``
SetObjectState ``Mandellbrot``, OFF
EndScript
EndObject
TextMenu ``RES2``
Definition
AttachToOBJECT, ``Resolution``
Font ``topaz``, 8; FontName, PointSize
PrintStyle PLAIN, 2, 3; Style, Pen1, Pen2
TextColors 0, 1, NORMAL; PenA, PenB, DrawMode
Text `` 2 ``
MenuFlags CHECKABLE
Highlight COMPLEMENT
ShortCutKey ``6``
EndScript
Occurred
Let Resolution=2
SetObjectState ``RES1``, OFF
SetObjectState ``RES4``, OFF
SetObjectState ``RES8``, OFF
EndScript
EndObject
TextMenu ``RES4``
Definition
AttachToOBJECT, ``Resolution``
Font ``topaz``, 8; FontName, PointSize
PrintStyle PLAIN, 2, 3; Style, Pen1, Pen2
TextColors 0, 1, NORMAL; PenA, PenB, DrawMode
Text `` 4 ``
MenuFlags CHECKABLE
Highlight COMPLEMENT
ShortCutKey ``7``

```

```

EndScript
Occurred
Let Resolution=4
SetObjectState ``RES1``, OFF
SetObjectState ``RES2``, OFF
SetObjectState ``RES8``, OFF
EndScript
EndObject
TextMenu ``RES8``
Definition
AttachToOBJECT, ``Resolution``
Font ``topaz``, 8; FontName, PointSize
PrintStyle PLAIN, 2, 3; Style, Pen1, Pen2
TextColors 0, 1, NORMAL; PenA, PenB, DrawMode
Text `` 8 ``
MenuFlags CHECKABLE
Highlight COMPLEMENT
ShortCutKey ``8``
EndScript
Occurred
Let Resolution=8
SetObjectState ``RES1``, OFF
SetObjectState ``RES2``, OFF
SetObjectState ``RES4``, OFF
EndScript
EndObject
TextMenu ``Default``
Definition
AttachToMENU, ``PlotInfo``
Font ``topaz``, 8; FontName, PointSize
PrintStyle PLAIN, 2, 3; Style, Pen1, Pen2
TextColors 0, 1, NORMAL; PenA, PenB, DrawMode
Text ``Default Settings``
MenuFlags NONE
Highlight COMPLEMENT
ShortCutKey ``D``
EndScript
Occurred
GotoCard ``MSettings``
EndScript
EndObject
*EndOfCard ``MSettings``
*****

```

Listing 2

Listing 2. CanDoMUC.hHeader File

```

/*-----INCLUDES-----*/
#include <proto/exec.h>
#include <proto/intuition.h>
#include <proto/graphics.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <float.h>

/*-----DEFS-----*/
#define MANDELBROT 0
#define JULIA 1

/*-----StructurePointerDeclarations-----*/
struct IntuitionBase *IntuitionBase;
struct GfxBase *GfxBase;

/*-----StructureDefinitions-----*/
struct complex{
    DOUBLE r;
    DOUBLE i;
};

struct ComplexRange{
    DOUBLE rmin, rmax;
    DOUBLE imin, imax;
};

/*-----GENERALFUNCTIONS-----*/
LONG OpenLibraries(void)
{
    IntuitionBase=(struct IntuitionBase *)
    OpenLibrary(``intuition.library``, 33);
    if (!IntuitionBase) return FALSE;

    GfxBase=(struct GfxBase *) OpenLibrary(``graphics.library``, 33);

```

```

if (!GfxBase) return FALSE;

return TRUE;
} /* OpenLibraries */

void CloseLibraries (void)
{
if (IntuitionBase) CloseLibrary((struct Library*)IntuitionBase);
if (GfxBase) CloseLibrary((struct Library*)GfxBase);
} /* CloseLibraries () */

```

Listing 3

Listing 3. CanDoMJC.c Program

```

/*-----INCLUDE HEADERS-----*/
#include 'CanDoMJC.h'

/*-----SUPERFUNCTIONS-----*/

void MPlot (struct Window *w, WORD maxdwell, LONG scale,
            struct ComplexRange *cr, struct complex *jc, UBYTE type)
{
    struct RastPort *rp;
    struct complex c, z, zloop;
    struct IntuiMessage *message;
    WORD xstart=0, ystart=0, x, y, xend, yend;
    WORD count;
    DOUBLE rstep, istep, zrtemp;
    ULONG oldIDCMPFlags;

    rp = w->RPort;
    oldIDCMPFlags = w->IDCMPFlags;
    ModifyIDCMP(w, MOUSEBUTTONS);

    xend = 8*(rp->BitMap->BytesPerRow)-1;
    yend = (rp->BitMap->Rows)-1;
    rstep = (cr->rmax - cr->rmin)*scale/xend;
    istep = (cr->imax - cr->imin)*scale/yend;

    SetRast(rp, 0);
    if (type == MANDELBROT) {
        for (c.i = cr->imax, y = ystart; y <= yend; c.i -= istep, y += scale) {
            if (message = (struct IntuiMessage *) GetMsg(w->UserPart)) {
                ReplyMsg((struct Message *) message);
                break;
            } /* if */
            for (c.r = cr->rmin, x = xstart; x <= xend; c.r += rstep,
                x += scale) {
                zloop.r = zloop.i = 0;
                count = 0;

                do {
                    zrtemp = zloop.r*zloop.r - zloop.i*zloop.i + c.r;
                    zloop.i = 2.*zloop.r*zloop.i + c.i;
                    zloop.r = zrtemp;
                    if ((zloop.r*zloop.r + zloop.i*zloop.i) > 4.) break;
                    ++count;
                } while (count < maxdwell);

                SetAPen(rp, count);
                if (scale == 1)
                    WritePixel(rp, x, y);
                else
                    RectFill(rp, x, y, x+scale-1, y+scale-1);
            } /* for */
        } /* if */
    }
    else if (type == JULIA) {
        for (z.i = cr->imax, y = ystart; y <= yend; z.i -= istep, y += scale) {
            if (message = (struct IntuiMessage *) GetMsg(w->UserPart)) {
                ReplyMsg((struct Message *) message);
                break;
            } /* if */
            for (z.r = cr->rmin, x = xstart; x <= xend; z.r += rstep, x += scale) {
                zloop.r = z.r;
                zloop.i = z.i;
                count = 0;

                do {
                    if ((zloop.r*zloop.r + zloop.i*zloop.i) > 4.) break;
                    ++count;
                    zrtemp = zloop.r*zloop.r - zloop.i*zloop.i + jc->r;
                    zloop.i = 2.*zloop.r*zloop.i + jc->i;
                    zloop.r = zrtemp;

```

```

} while (count < maxdwell);

                SetAPen(rp, count);
                if (scale == 1)
                    WritePixel(rp, x, y);
                else
                    RectFill(rp, x, y, x+scale-1, y+scale-1);
            } /* for */
        } /* else if */

        ModifyIDCMP(w, oldIDCMPFlags);
    } /* MPlot */

void SetComplexRange(struct ComplexRange *r, DOUBLE rmin,
                    DOUBLE rmax, DOUBLE imin, DOUBLE imax)
{
    r->rmin = rmin;
    r->rmax = rmax;
    r->imin = imin;
    r->imax = imax;
} /* SetComplexRange */

/*-----MAIN PROGRAM-----*/

LONG main (int argc, char *argv[])
{
    /* LOCAL VARIABLES */

    struct Window *CanDoWin;
    struct ComplexRange range;
    struct ComplexRange jrange;
    struct complex jc;

    char **dummy = NULL;
    char MandOrJulia;

    WORD maxdwell;
    WORD resolution;

    DOUBLE rmin, rmax, imin, imax;

    /* OPEN LIBRARIES */

    if (!OpenLibraries()) {
        CloseLibraries();
        return 1L;
    } /* if */

    /* EXECUTE PROGRAM IF RIGHT NUMBER OF ARGUMENTS */

    if (argc >= 9) {
        CanDoWin = (struct Window *) (strtol(argv[1], dummy, 10));
        maxdwell = (WORD) (strtol(argv[2], dummy, 10));
        resolution = (WORD) (strtol(argv[3], dummy, 10));
        rmin = strtod(argv[4], dummy);
        rmax = strtod(argv[5], dummy);
        imin = strtod(argv[6], dummy);
        imax = strtod(argv[7], dummy);
        MandOrJulia = *(argv[8]);

        if (MandOrJulia == 'M') {
            SetComplexRange(&range, rmin, rmax, imin, imax);
            MPlot(CanDoWin, maxdwell, resolution, &range, NULL, MANDELBROT);
        } /* if */

        else if (MandOrJulia == 'J') {
            jc.r = strtod(argv[9], dummy);
            jc.i = strtod(argv[10], dummy);
            SetComplexRange(&jrange, rmin, rmax, imin, imax);
            MPlot(CanDoWin, maxdwell, resolution, &jrange, &jc, JULIA);
        } /* else if */
    } /* if argc >= 2 */

    CloseLibraries();
} /* main */

```

·AC·

Please Write to:
 Randy Finch
 c/o Amazing Computing
 P.O. Box 2140
 Fall River, MA 02722-2140