gtk-fortran: a GTK / Fortran binding

1. Vincent MAGNIN, 2. James TAPPIN,
3. Jens HUNGER, 4. Jerry DELISLE

1. Lille University (France), 2. RAL Space (UK),
3. Technische Universität Dresden (Germany), 4. GFortran Team (USA)

FortranCon 2020, Zurich, 2-4 July 2020

Plan

The scientist and the unknown pleasures of visualization...

- Do scientists dream of ~~electric sheep~~ being artists?
- Scientific visualization is essential and is an art (left).
- Nothing in the Fortran standard, except ASCII art? (right)

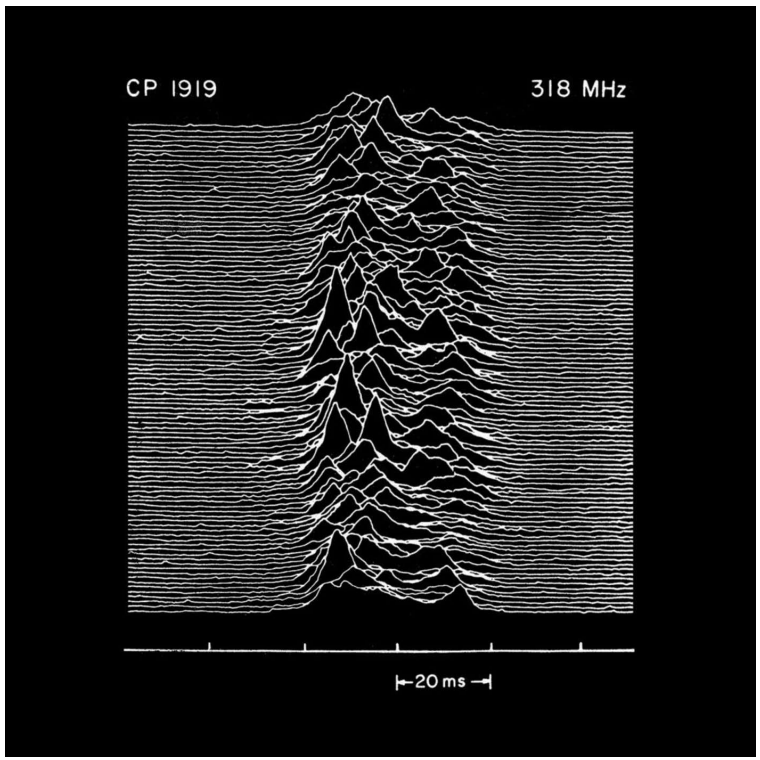

Figure: first discovered pulsar [1].

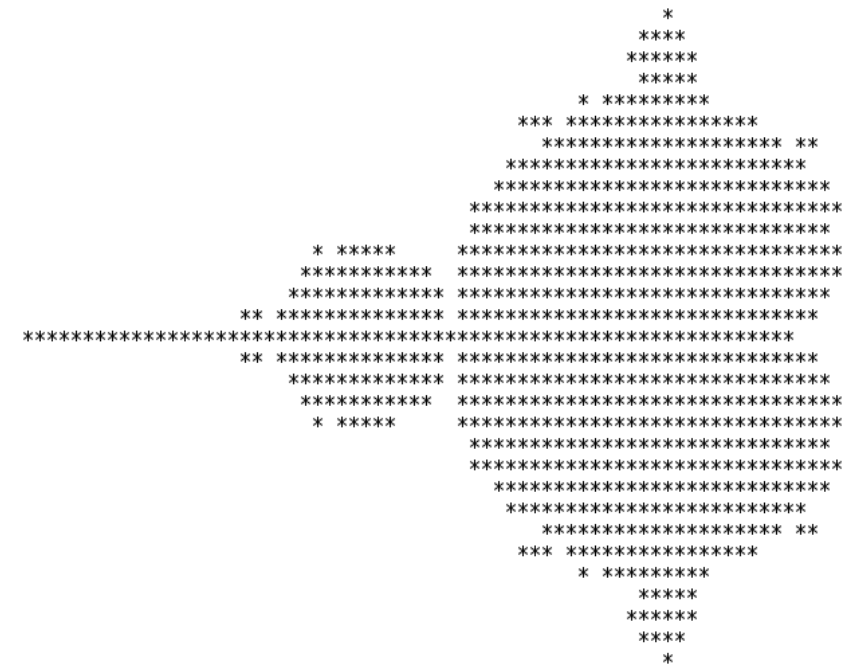

Figure: the Mandelbrot set in 1978.

You can't always get what you want

- Visual Fortran (Digital, Compaq, Intel) has the Quickwin library (Windows).
- When migrating to Linux, I was looking for something:
  - to make bitmap (or vectorial) drawing,
  - and scientific plotting.
  - To build *Graphical User Interfaces* (GUI).
  - Using only standard Fortran.
  - Multi-platform: Linux, Windows, MacOS...
  - Free software (as in *free* beer and *free*dom).
  - A perennial solution.
- There was/is a lot of great tools like pilib (GTK 2, but needing a C compiler), DISLIN, PLplot, gnuplot, interfacing with another language (japi), etc.
- But I still hadn't found what I was looking for...

But if you try sometime you find: GTK

2nd May 2007 13:10

Tobias BURNUS posted on the `comp.lang.fortran` newsgroup a demo creating an empty GTK 2 window:

- GTK is mainly written in C language,
- and Fortran 2003 introduced interoperability with C.

About GTK

- GNU LGPL license.
- Multi-platform.
- Who's behind? The GNOME Foundation (RedHat...).
- 2002: GTK 2.0
- 2011: GTK 3.0
- 2020(?): GTK 4.0

GTK is not a library...

It's a collection of libraries !

- GLib: general-purpose utility library,
- GObject: the GLib Object System,
- GTK itself,
- GdkPixbuf: pixel drawing (even without GUI),
- Cairo: vectorial drawing,
- GIO : input/output,
- GDK, Pango, ATK...

A GTK / Fortran binding

gtk-fortran

- Project launched with J. DELISLE, J. TAPPIN and J. HUNGER.
- First commit: 9 January 2011.
- Now $\approx 10\,000$ functions in gtk-fortran!
- Not only about GUI and graphics! But also strings, regex, date and time, files, network utilities, lists, trees, hash tables, etc.
- Only 500 functions used in our tools and examples...

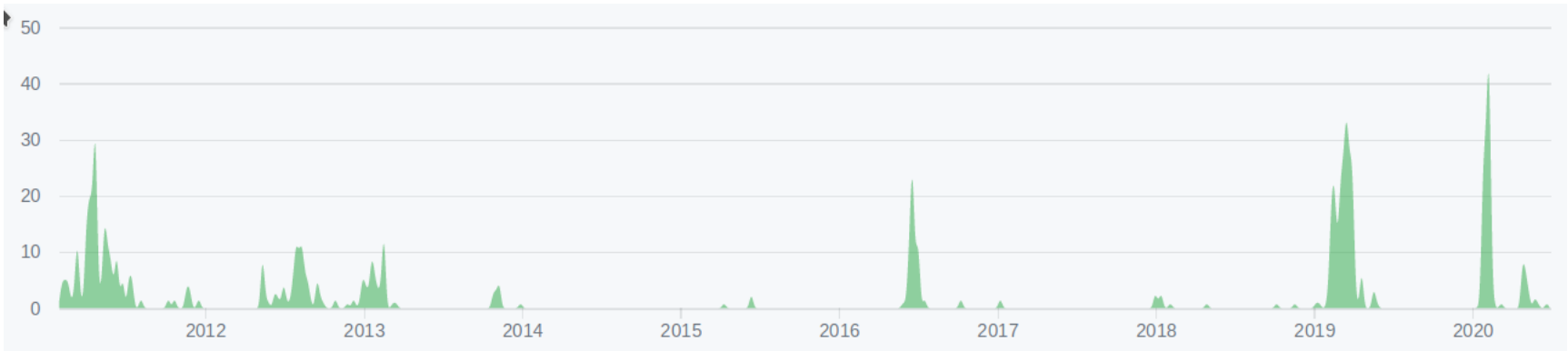

Figure: Commits in the GTK 3 branch.

Installing and building gtk-fortran

- Download the zip from https://github.com/vmagnin/gtk-fortran/wiki or git clone:

```
$ git clone git@github.com:vmagnin/gtk-fortran.git
```

- Install tools and dependencies (Debian/Ubuntu names):

```
$ sudo apt install gfortran cmake libgtk-3-dev
    libplplot-dev plplot-driver-cairo
```

- Build the project and install the library:

```
$ mkdir build && cd build
$ cmake ..
$ make
$ sudo make install
```

- Compile your own program:

```
$ gfortran my_app.f90 $(pkg-config --cflags --libs
    gtk-3-fortran)
```

And I went down to the demonstration

Figure: hl_cairo_clock.f90 (Kubuntu)

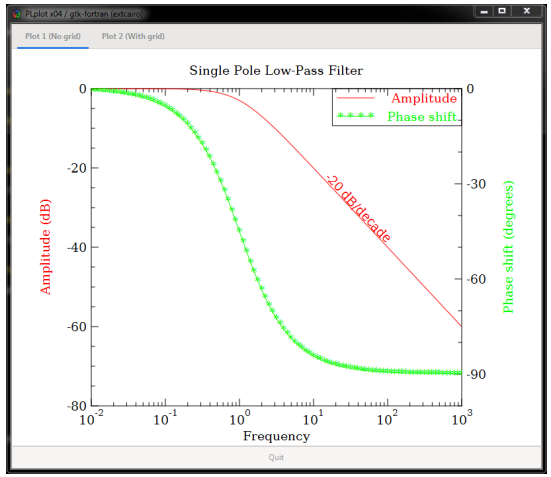

Figure: hl_plplot4e.f90 (Windows 7)

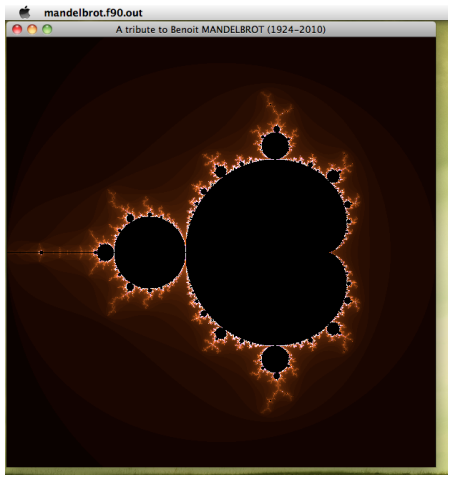

Figure: mandelbrot_pixbuf.f90 (MacOS)

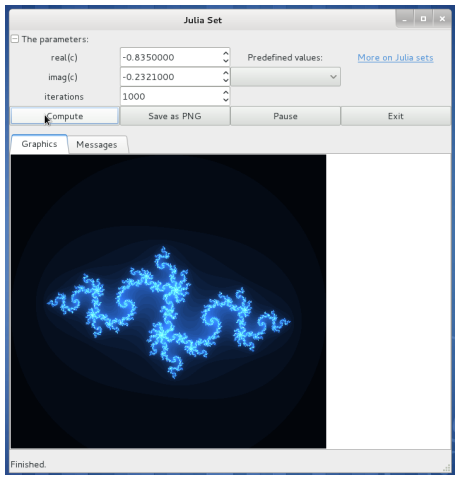

Figure: julia_pixbuf.f90 (Fedora)

Some projects using gtk-fortran

- "The VCU nuclear reactor simulator, Richmond Pile 3, is a classroom and research tool that emulates a large commercial pressurized water reactor (PWR)." [2]

  https://egr.vcu.edu/departments/mechanical/research/nuclear-simulator-lab/

- Uncertradio [3]:

  https://www.thuenen.de/de/fi/arbeitsbereiche/meeresumwelt/leitstelle-umweltradioaktivitaet-in-fisch/uncertradio/

Nobody said it was easy...

Event driven programming

- Based on a *main loop,* idle most of the time.
- If something happens a *signal* is emitted,
- received by the concerned *widget,*
- which calls its corresponding *callback function.*

It's all about C pointers and C types

- C pointers toward functions, objects, data...
- C types: int, double, char...
- The GTK official documentation is totally C oriented: to be able to read C function prototypes is better...

Example: the button widget

To create my "Bye" button, I need to call that C function whose prototype is in `/usr/include/gtk-3.0/gtk/gtkbutton.h`:

```
GtkWidget* gtk_button_new_with_label(const gchar *label);
```

The gtk-fortran interface for that function is in `src/gtk-auto.f90`:

```
function gtk_button_new_with_label(label) bind(c)
  use iso_c_binding, only: c_ptr, c_char
  type(c_ptr) :: gtk_button_new_with_label
  character(kind=c_char), dimension(*) :: label
end function
```

In my Fortran program:

```
  use iso_c_binding
  use gtk, only:  gtk_button_new_with_label, g_signal_connect

...
  my_button = gtk_button_new_with_label("Bye"//c_null_char)

  ! I need also to connect my button to a callback function:
  call g_signal_connect(my_button, "clicked"//c_null_char, c_funloc(
       my_callback_function))
```

How the Fortran interfaces are generated?

Before each gtk-fortran release

- The python program `cfwrapper.py` parses the GTK `.h` header files,
- generates the `src/*-auto.f90`,
- using especially regular expressions.

```
$ ./cfwrapper.py -g 3
 Pass 1: looking     enumerators, funptr and derived types...
 Pass 2: looking     C functions...
/usr/include/atk-1.0                =>  atk-auto.f90
    85781 bytes
/usr/include/cairo                  =>  cairo-auto.f90
    133740 bytes
/usr/include/gdk-pixbuf-2.0         =>  gdk-pixbuf-auto.f90
    58223 bytes
/usr/include/glib-2.0               =>  glib-auto.f90
    1299171 bytes
/usr/include/gtk-3.0/gdk            =>  gdk-auto.f90
    208504 bytes
/usr/include/gtk-3.0/gtk            =>  gtk-auto.f90
    1319554 bytes
/usr/include/gtk-3.0/unix-print     =>  unix-print-auto.f90
    24488 bytes
/usr/include/pango-1.0              =>  pango-auto.f90
```

Tools to ease the pain

A High Level library (J. Tappin)

- Offers a simplified interface for most useful GUI functions.
- Can be mixed with direct GTK calls.

gtkf-sketcher (J. Hunger)

- Complicated GUI can be graphically designed with Glade,
- saved in an XML file,
- used by gtkf-sketcher to generate a gtk-fortran program.

Other tools

- usemodules.pl: to generate your "USE, ONLY:" statements
- Some shell scripts.

How to contribute?

- Build and test with various systems and compilers:
  - use the gtk3 stable branch,
  - or the gtk4 branch if you are already familiar with GTK.
- Report issues on GitHub.
- Help improve the documentation.
- Write tutorials.
- Write new examples (only 500 functions tested).
- ...

SWOT matrix of the project

Strengths

- Standard Fortran.
- GTK is perenial and multi-platform.
- gtk-fortran maintained since 2011.
- Includes a High Level library.
- Many commented examples.

Weaknesses

- Few contributors.
- No GTK programming tutorial.
- The cfwrapper.py needs refactoring.
- The binding is not 100 % complete.

Opportunities

- FortranCon 2020.
- Fortran-lang Community and tools.
- New compilers: Flang, LFortran...
- GTK 4 soon released (Fall 2020?).
- Use GObject Introspection?

Threats

- Fortran is losing users.

Thank you for listening!

I will be pleased to answer your questions now

or later on Slack, GitHub, Fortran-lang Discourse...

Acknowledgments

- Tobias BURNUS who posted on the `comp.lang.fortran` newsgroup (2nd May 2007 13:10) an example of a program creating an empty GTK+ 2 window.
- Contributors and users who are not in the authors: Mehdi CHINOUNE (CMake parallelisation), Ian HARVEY...

Useful links

- The gtk-fortran home [4]:
  https://github.com/vmagnin/gtk-fortran/wiki
- The GTK home:
  https://www.gtk.org/
- Where to find its official documentation:
  https://developer.gnome.org/
- The GTK developers blog:
  https://blog.gtk.org/
- The GNOME Discourse:
  https://discourse.gnome.org/c/platform
- Fresh GTK News:
  https://gitlab.gnome.org/GNOME/gtk/-/blob/master/NEWS

References

Harold Dumont Craft.
Radio observations of the pulse profiles and dispersion measures of twelve pulsars.
Ithaca, N.Y., 1970.
Ph.D. Thesis, OCLC: 741854775.

D. Tincher, S. Bilbao y León, and J. Miller.
Progress on real-time, interactive modeling and simulation of nuclear power plants.
volume 2017-September, 2017.

Günter Kanisch.
Generalized evaluation of environmental radioactivity measurements with uncertradio. part i: Methods without linear unfolding.
Applied Radiation and Isotopes, 110:28 – 41, 2016.

V. MAGNIN, J. TAPPIN, J. HUNGER, and J. DELISLE.
gtk-fortran: a GTK+ binding to build Graphical User Interfaces in Fortran.
The Journal of Open Source Software, 5(34), February 2019.