New ways to use FreeFem++

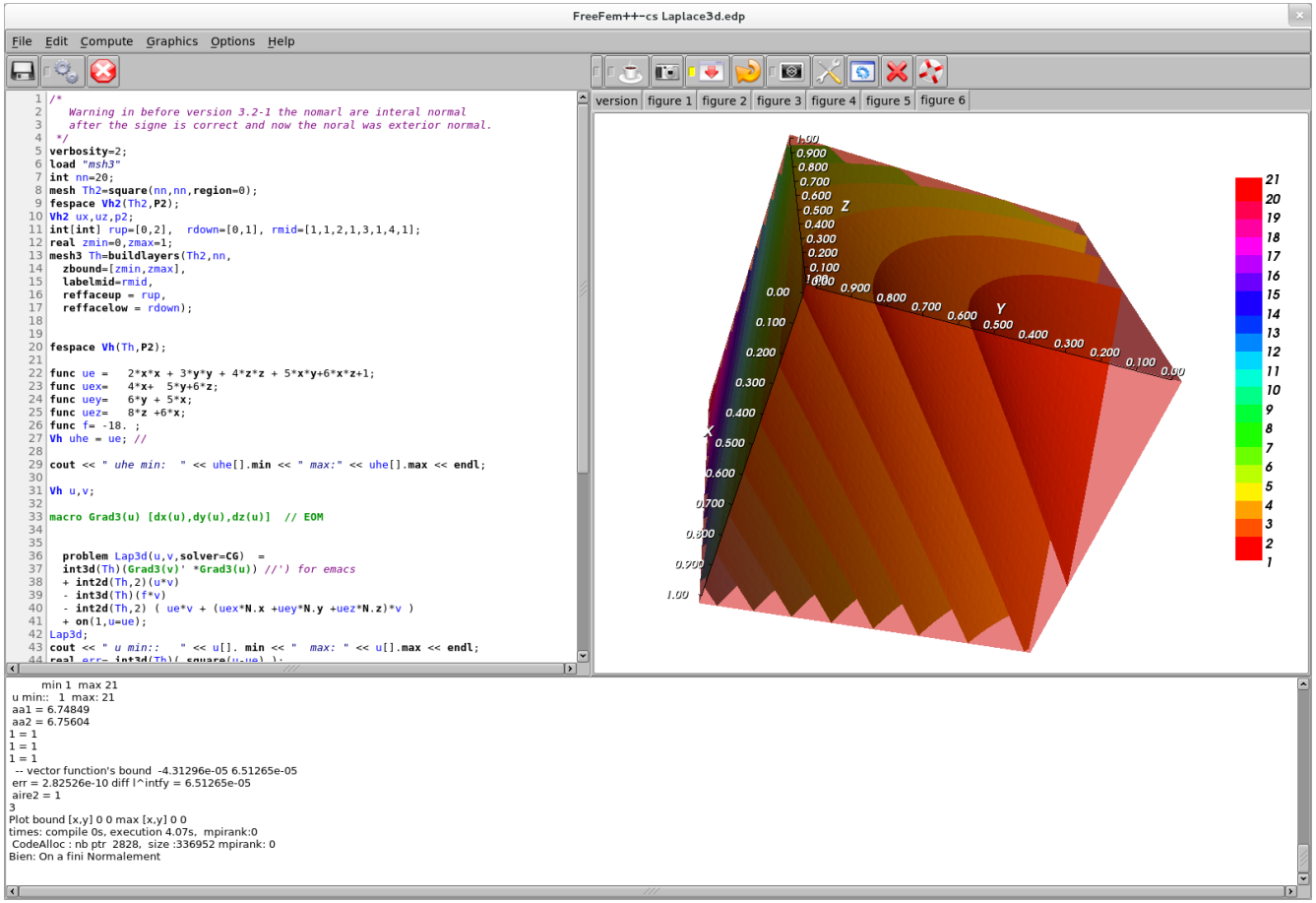
Antoine Le Hyaric

Laboratoire Jacques-Louis Lions
Université Pierre et Marie Curie

Antoine.Le_Hyaric@upmc.fr

December 15, 2015

Introduction to FreeFem++-cs



Download : https://www.ljll.math.upmc.fr/lehyaric/ffcs

Tutorial : https://hal.archives-ouvertes.fr/hal-01169630

FreeFem++-cs uses the FreeFem++ package

Implementation

- Reused fflang.cpp without FF library, ffexe.cpp calls FF executable directly
- Replaced TCP socket communication with plain files

Current Limitations

- All output is buffered until the end of the script
- Requires a recent version of FreeFem++ (which accepts the -nowait option) (cf src/Graphics/getprog-unix.hpp)

Long-term Advantages

- Always up-to-date with FreeFem++ versions
- One Windows version for both Win32/Win64

Live example

Current Issues

Glitches

- Need to download FF separately
- Reduced number of precompiled versions
- MacOS version delayed

Improvements

- Improved Windows stability
- Upgraded to latest versions of FLTK and VTK
- added cross-references in FreeFem++ and FreeFem++-cs

All news available as usual at www.ljll.math.upmc.fr/lehyaric/ffcs and in the NEWS file.

MacOS-specific Issue

MacOS has been updated again, to version 10.11. We need to find a common environment where FLTK, VTK, OpenGL work :

- clang+Cocoa : NSview is not shared between FLTK and VTK both expect to create their own NSView instance
- MacPorts+X11 (i.e. "Linux under XQuartz") : current VTK/MacOS is hardwired for platform-dependant OpenGL calls and does not compile cleanly with GNU g++

Options :

- patch FLTK
- patch VTK
- replace FLTK with another GUI toolkit

Javascript

- Established language (2015 Stack Overflow survey)
- Runs in the Javascript VM, contained in a web page
- GUI frameworks available (jQuery, Bootstrap, etc) (example)
- Works with most browsers (Firefox, Safari, Internet Explorer/Edge, etc)

Benefits of Javascript for FreeFem++

- Only a limited rewrite (thanks to Emscripten)
- Well-known easy-to-edit file format : HTML
- Many available tools
  (MathJax, editors, 2D and 3D graphics, etc)
- Portable across Windows, Linux, MacOS, Android, iOS, etc
- Most configuration issues already solved by the VM
- No installation required
- Nearly everyone has a smartphone nowadays

Conversion to Javascript, 1^{st} step : translating C++

1^{st} step : Translating FreeFem++ from C++ to Node.js with Emscripten

```
int main(int argc,char *argv[]){

  // mount local file system
  EM_ASM(
         FS.mkdir("/work");
         FS.mount(NODEFS,{root:"."},"/work");
         );

  // change path to script file relative to the mounted file system
  string f=argv[1];
  f="/work/"+f;

  // call FreeFem++
  freefem(f.c_str());
  return 0;
}
```

The corresponding source file is node.cpp

2^{nd} step : connecting with the HTML DOM

- standard HTML
- One-window Panel-oriented based layout like regular FreeFem++-cs

```
<!-- script text -->
<textarea class="ffjs">
mesh Th=square(10,10);
plot(Th);
</textarea>

<!-- run button -->
<button type="button" onclick="ffjs_evaluate();" >Run</button>

<!-- textual output -->
<div id="ffjs_stderr" style="font-family:monospace;font-size:small;"></div>
<div id="ffjs_stdout" style="font-family:monospace;font-size:small;"></div>

<!-- 2D graphical output -->
<canvas class="ffjs_graph" style="width:100%;" data-ffjs="default"></canvas>
```

3^{rd} step : HTML5 Graphics

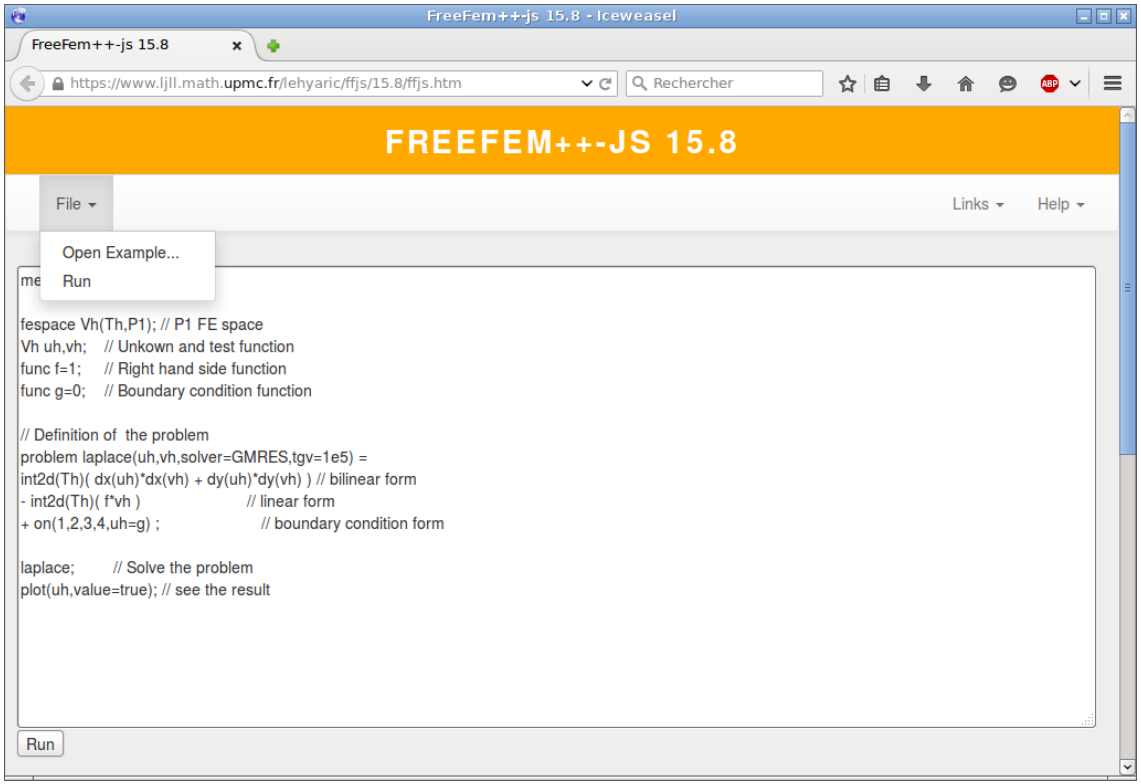
Implement HTML5 canvas calls in src/Graphics/sansrgraph.cpp

```
...

void rmoveto(reel x, reel y)
{
  currx = scalx(x);
  curry = scaly(y);

#ifdef FFJS_GRAPH
  EM_ASM_INT({ffjs_rmoveto($0,$1)},currx,curry);
#endif
}

...
```

What can we do with the Javascript FreeFem++?

- Javacript FreeFem++ can run in any web page
- The web page design is independent of FreeFem++
- There are many elements (HTML, open source editors and graphics, etc) available in " "
- Some potentially interesting designs :
  - Web-based IDE
  - Wiki
  - Literate programs
  - Programs which run on any platform

Web-based GUI



Live example

(Also accessible online : `http://www.ljll.math.upmc.fr/lehyaric/ffjs/ffjs.htm`)

Wiki

Mixing together FreeFem++, Javascript and a Wiki engine

Online programs which can be written concurrently, wiki-style

www.um.es/freefemv3/ff++/pmwiki.php?n=Didactic.Didactic

Requirements :

- Internet server
- Recent software (eg PmWiki 2.2.58) to view graphical results

Literate Programming

In summary, a single HTML page may contain :

- regular text
- equations (MathJax)
- interactive FreeFem++ commands

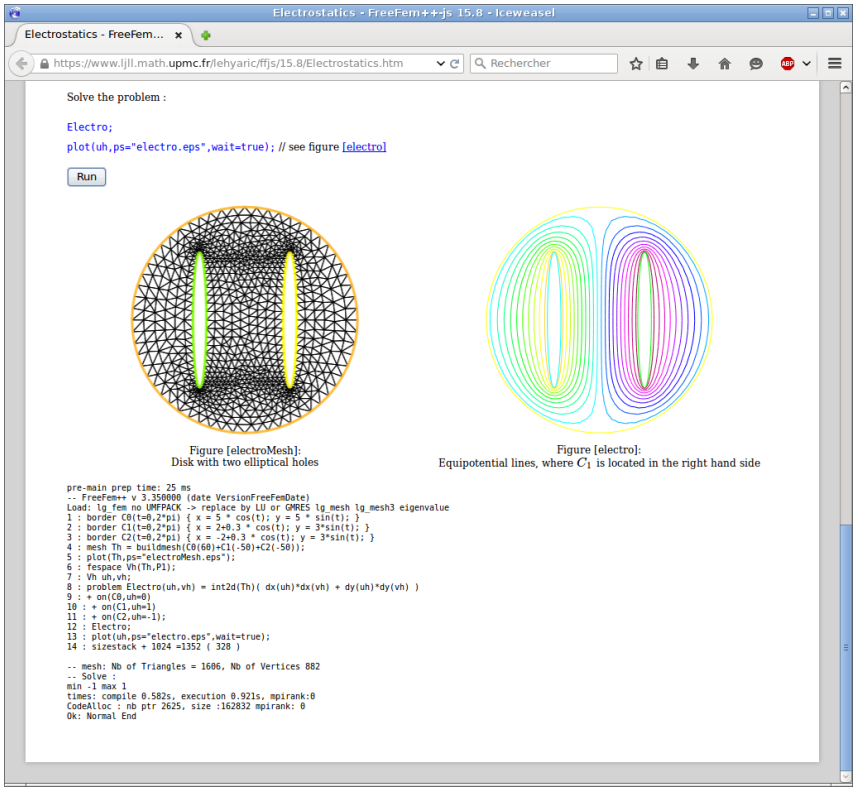
Live example

Also accessible online :

http://www.ljll.math.upmc.fr/lehyaric/ffjs/template.htm

Complete Example

Translated from http://www.um.es/freefemv3/ff++/pmwiki.php?n=FFDoc.Electrostatics



Result in http://www.ljll.math.upmc.fr/lehyaric/ffjs/Electrostatics.htm

Going full circle : Back to a terminal

The Node.js FreeFem++ binary can be run from a terminal, like the regular C++-compiled package. All the Node.js platforms become automatically available to FreeFem++.

Hardware-dependant FreeFem++ :

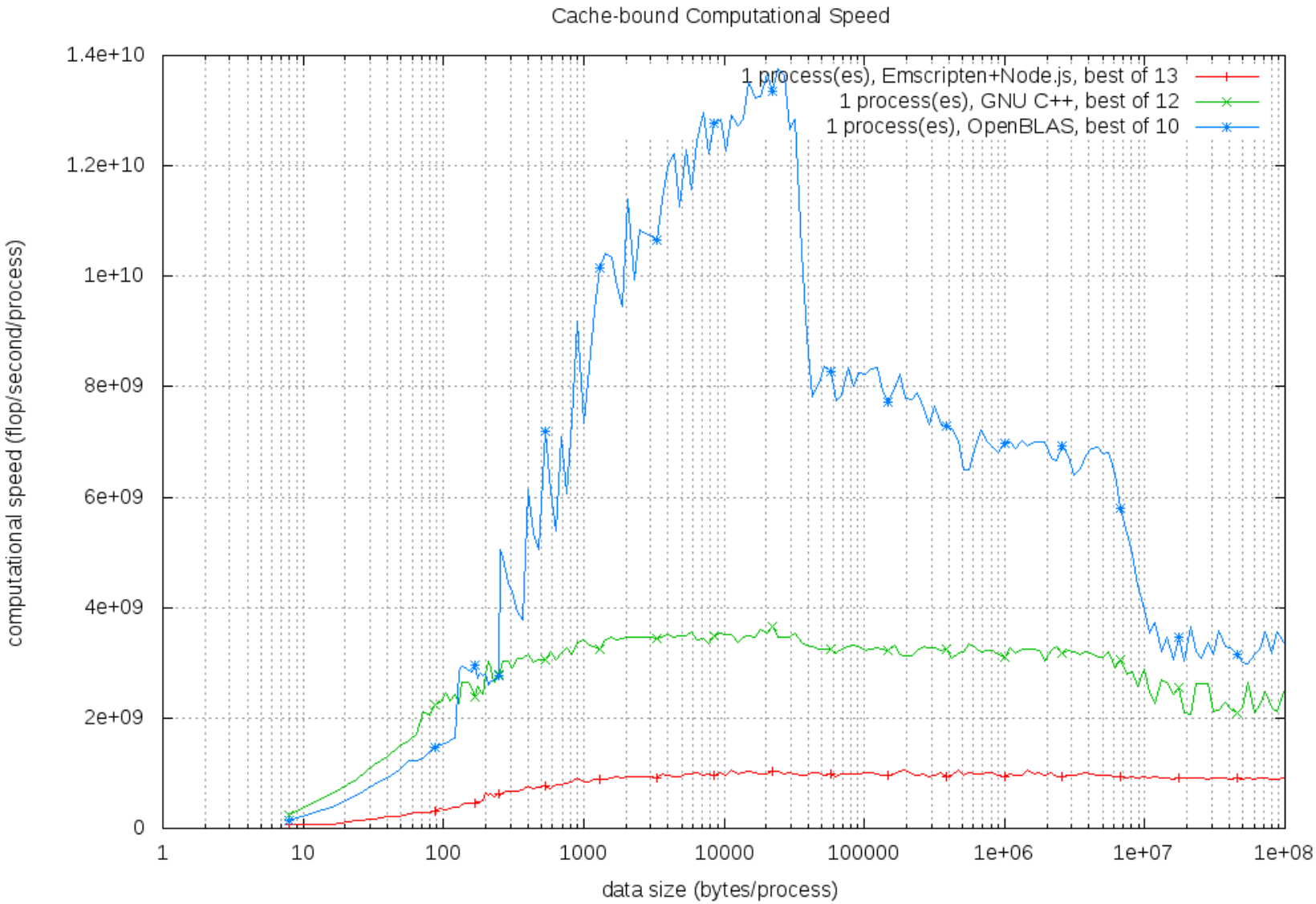
```
> FreeFem++ examples++-tutorial/Laplace.edp
```

Hardware-independant FreeFem++ :

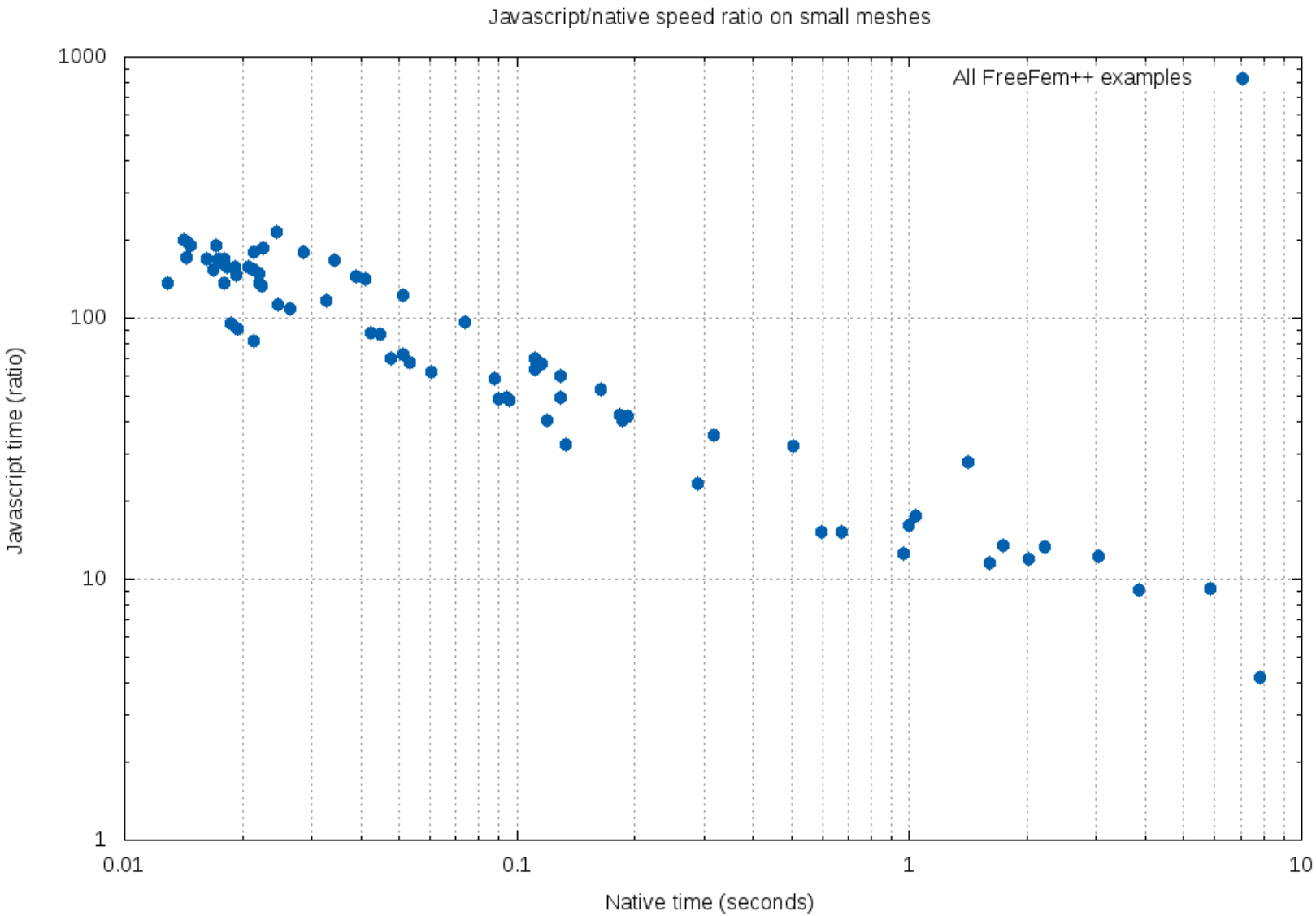
```
> nodejs nodefreefem.js examples++-tutorial/Laplace.edp
```

What is the speed difference between the two options?

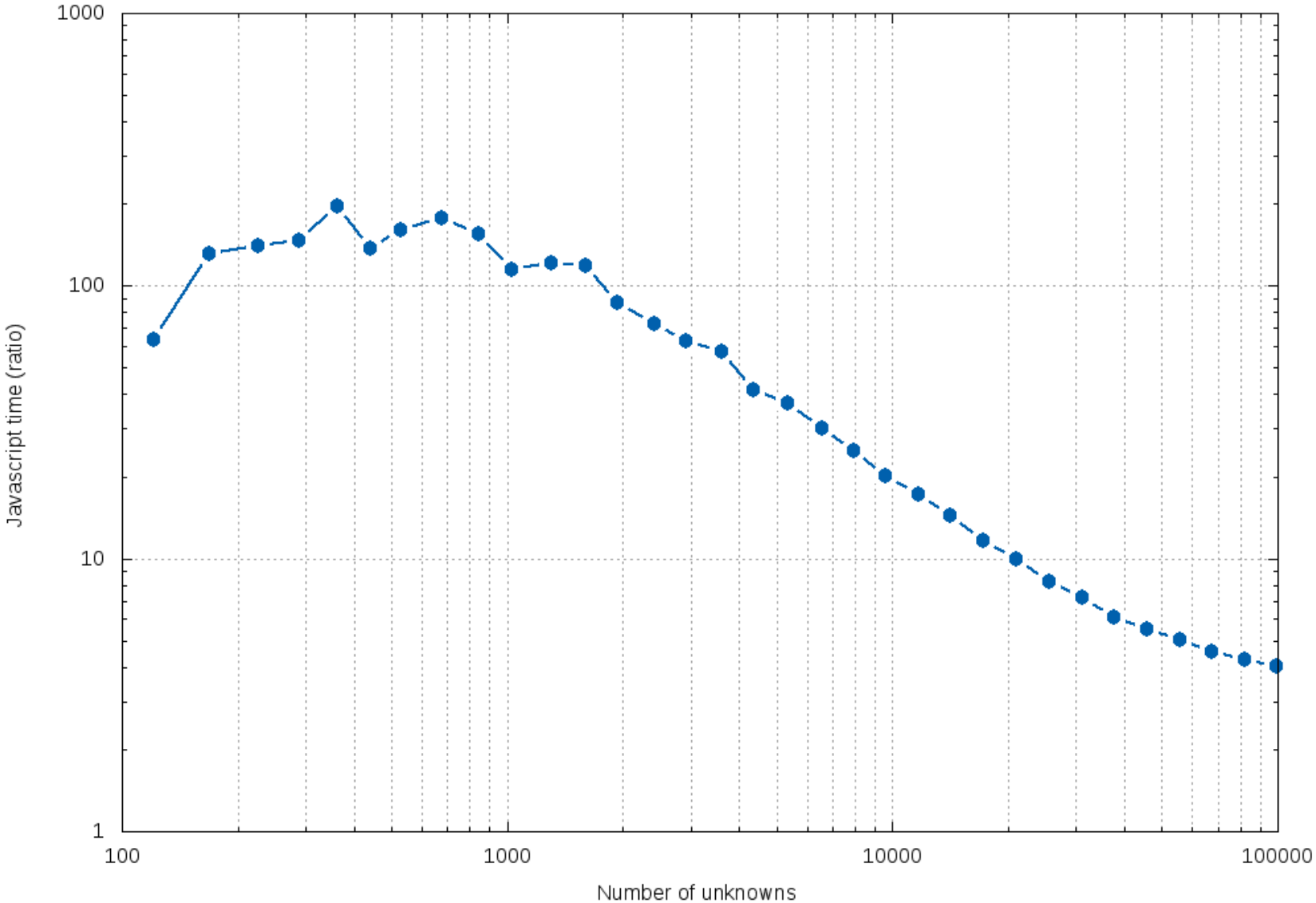
Javascript speed results



Speed ratio for existing examples



Speed ratio depending on problem size



Conclusion and Future Work

- Develop the new designs
- Parallelise with “web-workers”
- Connect FreeFem++cs server (for speed) and FreeFem++-js (for portability)