Rust Wrapper for MFEM

Máté Kovács @ MFEM Community Workshop
October 22, 2024
LLNL

About Me

Hi, I'm Máté.

- pronounced as in yerba maté
- C++ enthusiast since 2002
- Rust fanboy since 2016
- MFEM dabbler since 2021
- organizer of the Tokyo Rust community since 2022

  tokyorust.org

Rust Wrapper for MFEM

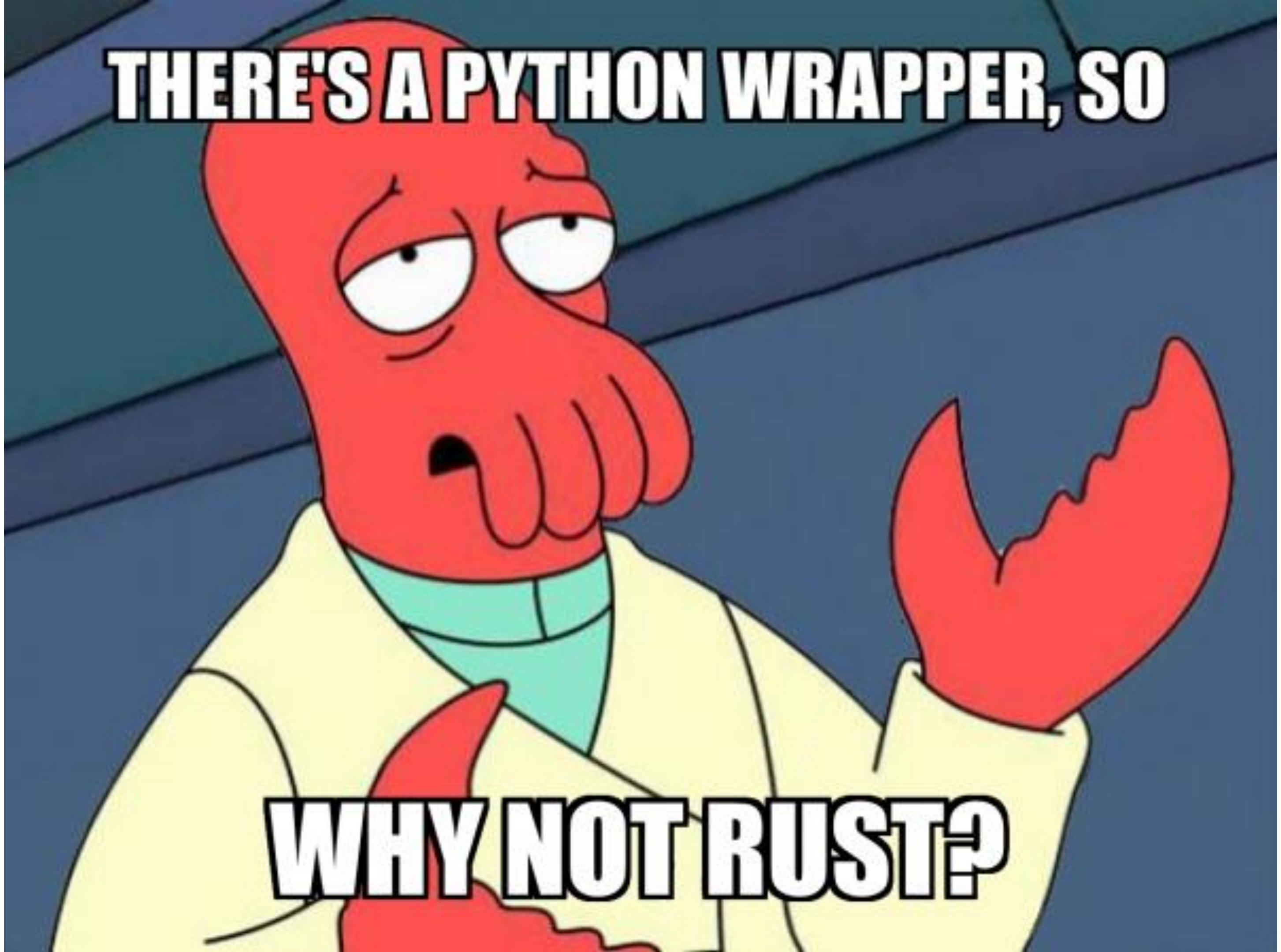
Key Points

- How MFEM would benefit from a Rust wrapper.
- Quick overview of the wrapper I started building.
- Call to Action: Please help build and maintain it!

Benefits of a Rust Wrapper

Rust: Benefits for the MFEM Community

- easier to use, lower barrier to entry
  - more compile-time checks to guide correct use
  - universal, user-friendly package manager & build tool
- equivalent to C++ in terms of performance
  - robust, low-overhead integration with C++
- an incoming wave of adopters on the horizon
  - DARPA project to auto-convert C code to Rust
  - White House press release on memory-safe languages


THERE'S A PYTHON WRAPPER, SO
WHY NOT RUST?

The Wrapper: MFEM-rs

MFEM-rs: Summary

- already available on the Rust community's registry: https://crates.io/crates/mfem
- has automatically generated documentation https://docs.rs/mfem
- you only need a single line of configuration to depend on MFEM
- choose to link against on-system MFEM or build from source
- build and run your code with a single command
- use MFEM from idiomatic Rust code, have the compiler check it

Compile-Time Checks

Example: MFEM App in C++

```
using namespace mfem;

GridFunction make_gridfunc(Mesh  &mesh) {
    H1_FECollection fec(1, mesh.Dimension());
    FiniteElementSpace fespace(&mesh, &fec);
    return GridFunction(&fespace);
}

int main(int argc, char *argv[]) {
    Mesh mesh("../data/star.mesh");
    GridFunction x = make_gridfunc(mesh);
    ConstantCoefficient one(1.0);
    x.ProjectCoefficient(one);
}
```

Looks okay,
I guess..? 🤔

Example: MFEM App in C++

```
$ make example_mfem_app

g++  -O3 -std=c++11 -I..  example_mfem_app.cpp -o
example_mfem_app -L.. -lmfem
```

So far so good! 🤠

Example: MFEM App in C++

```
$ ./example_mfem_app

zsh: segmentation fault  ./example_mfem_app
```

Whoops! 🤯

Let's write the same app in Rust!

Example: MFEM App in Rust

```rust
use mfem::*;

fn make_gridfunc(mesh: &Mesh) -> GridFunction {
    let fec = H1FeCollection::new(1, mesh.dimension(), BasisType::GaussLobatto);
    let fespace = FiniteElementSpace::new(&mesh, &fec,  1,
OrderingType::byNODES);
    return GridFunction::new(&fespace);
}

fn main() {
    let mesh = Mesh::from_file("data/star.mesh").expect("Failed to load mesh");
    let x = make_gridfunc(&mesh);
    let one = ConstantCoefficient::new(1.0);
    x.project_coefficient(&one);
}
```

Example: MFEM App in Rust

```
$ cargo build

error[E0515]: cannot return value referencing local variable `fespace`
 --> src/main.rs:6:12
  |
6 |     return GridFunction::new(&fespace);
  |            ^^^^^^^^^^^^^^^^^^--------^
  |            |                 |
  |            |                 `fespace` is borrowed here
  |            returns a value referencing data owned by the current function
error[E0515]: cannot return value referencing local variable `fec`
 --> src/main.rs:6:12
  |
5 |     let fespace = FiniteElementSpace::new(&mesh, &fec, 1, OrderingType::byNODES);
  |                                                  ---- `fec` is borrowed here
6 |     return GridFunction::new(&fespace);
  |            ^^^^^^^^^^^^^^^^^^^^^^^^^^^ returns a value referencing data owned by the
current function
For more information about this error, try `rustc --explain E0515`.
error: could not compile `example_mfem_app` (bin "example_mfem_app") due to 2 previous
errors
```

Package Manager & Build Tool

C++: The Package Manager & Build Tool Situation

Rust: Cargo is Universal & User-Friendly

"The best thing about Cargo is not that it's the best build system [..],

but that there's nothing else for Rust."

https://news.ycombinator.com/item?id=24846876 – Kornel Lesiński

Community infrastructure built around Cargo:

- https://crates.io
- https://docs.rs
- https://lib.rs

Let's set up an MFEM app in Rust!

Example: Setting Up an MFEM App in Rust

```
$ mkdir example_mfem_app && cd example_mfem_app

$ cargo init

     Created binary (application) package

$ cargo add mfem

    Updating crates.io index
      Adding mfem v0.2.0 to dependencies.
             Features:
             + bundled
    Updating crates.io index
```

Example: Layout of an MFEM App in Rust

```
$ tree
.
├── Cargo.lock
├── Cargo.toml
└── src
    └── main.rs
```

```toml
# Contents of Cargo.toml

[package]
name = "example_mfem_app"
version = "0.1.0"
# Rust editions are like C++11, etc.
edition = "2021"

[dependencies]
# Tells Cargo to depend on MFEM
mfem = "0.2.0"
```

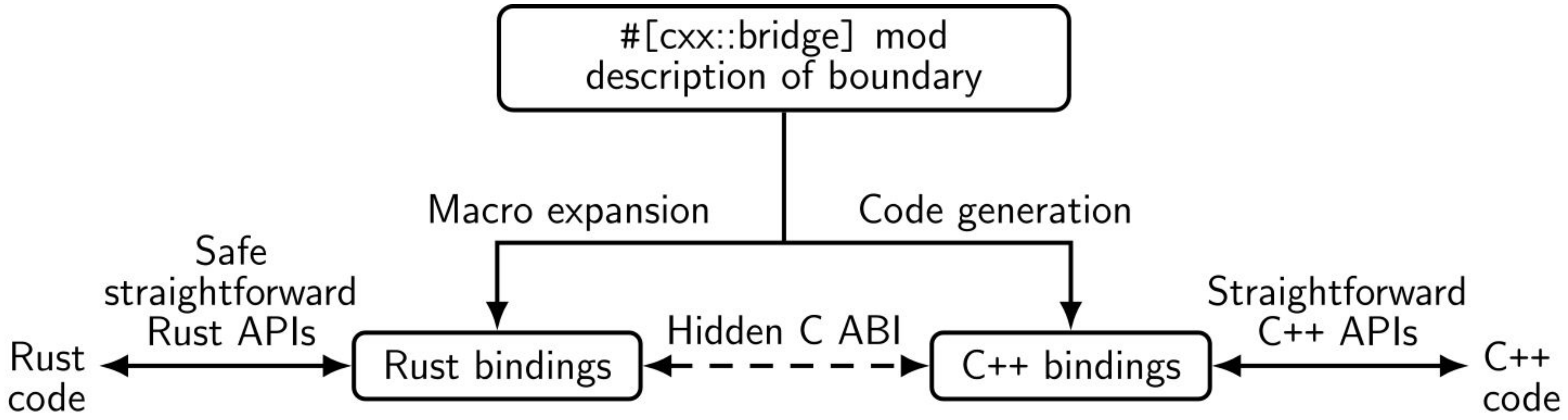
Performance

Performance: The Language C++ vs Rust

	Time [s]		Memory [bytes]		GZipped Source [bytes]	
Problem	C++	Rust	C++	Rust	C++	Rust
fannkuch-redux	**3.26**	3.88	**19,712**	19,804	1535	**1260**
n-body	**2.15**	2.19	**19,736**	19,804	1933	**1881**
spectral-norm	0.72	0.72	19,884	**19,804**	**1050**	1132
mandelbrot	**0.89**	0.94	34,944	**33,408**	1797	**1301**
pidigits	0.87	**0.71**	**19,736**	19,804	804	804
regex-redux	1.18	1.18	276,148	**154,096**	2856	**994**
fasta	0.78	**0.77**	**19,712**	19,804	2758	**2533**
k-nucleotide	**2.02**	2.84	156,512	**133,840**	1634	**1585**
reverse-complement	0.71	**0.53**	499,712	**498,816**	**2099**	2762
binary-trees	**0.96**	1.08	201,536	**198,656**	896	**771**

source: https://benchmarksgame-team.pages.debian.net/benchmarksgame/index.html

Performance: Binding Rust ❤️ C++



"The resulting FFI bridge operates at zero or negligible overhead, i.e. no copying, no serialization, no memory allocation, no runtime checks needed." – https://cxx.rs

Future Work

MFEM-rs: Future Work

There's a lot to do!

- Incrementally extend API coverage and translate all Examples.
- Help streamline the MFEM API.

  E.g. around `const` correctness
- Make it work with MPI (Message Passing Interface).

  https://crates.io/crates/mpi
- Cover other components (Hypre, etc.).

Please Help!

MFEM-rs: How You Can Help

- Try it out and give feedback; file issues.
- Streamline MFEM's C++ API.
- Contribute to the Rust wrapper itself.

Try out MFEM-rs

MFEM-rs: Try It Out & Give Feedback

- Get Rust from https://rustup.rs.
- Follow the steps from earlier to create an MFEM app in Rust.
- In `src/main.rs`, write some code using the `mfem` crate.
    - Read https://docs.rs/mfem to see what's available.
- If something doesn't work, or something you want isn't there:
    - File an issue on https://github.com/mkovaxx/mfem-rs.
    - Consider contributing the necessary changes.

Help streamline MFEM's C++ API

Context: Const Correctness in Rust

The idea of `const` correctness is more important in Rust than in C++.

Rust ensures that every piece of memory can have either multiple const references to it, or a single mutable reference, but never both at the same time. This simplifies concurrent code, among other things.

Consequences:

- Variables (and references) are immutable (aka. const) *by default*.
- Mutable variables (and refs) are marked with the `mut` keyword.

Help: Streamline MFEM's C++ API

The MFEM API often requires mutability in surprising places.

For example, look at the following signature:

```
void GridFunction::ProjectCoefficient(Coefficient& coeff);
```

From a semantics point of view, it seems that it should instead be:

```
void GridFunction::ProjectCoefficient(Coefficient const& coeff);
```

The actual signature isn't only startling, but also less usable from Rust.

Help: Streamline MFEM's C++ API

I guess `GridFunction::ProjectCoefficient()` is like that because `Coefficient::Eval()` is also not `const`.

I would hope that such nits could be cleaned up incrementally.

MFEM-rs currently has ugly hacks to get around these.
Likely unsafe… 😅

Contribute to MFEM-rs

Help: Contribute to MFEM-rs

- No time for details, but here's an example pull request:

  https://github.com/mkovaxx/mfem-rs/pull/1

Zooming Out

MFEM-rs: Zooming Out

Bring together modeling & simulation for Rust!

- OpenCascade: open-source B-rep kernel for modeling
  - Used in e.g. FreeCAD
  - Rust wrapper: opencascade-rs
- MFEM: needs no introduction here
  - Rust wrapper: mfem-rs

MFEM-rs: Zooming Out

- end-to-end modeling & simulation in a Rust app:
  - compute a shape (with OpenCascade)
  - test that shape (with MFEM)
- all in Rust, so you can
  - share functionality as Cargo crates (packages)
  - easily build on top of other people's work
- what's missing?
  - need a package for meshing B-reps

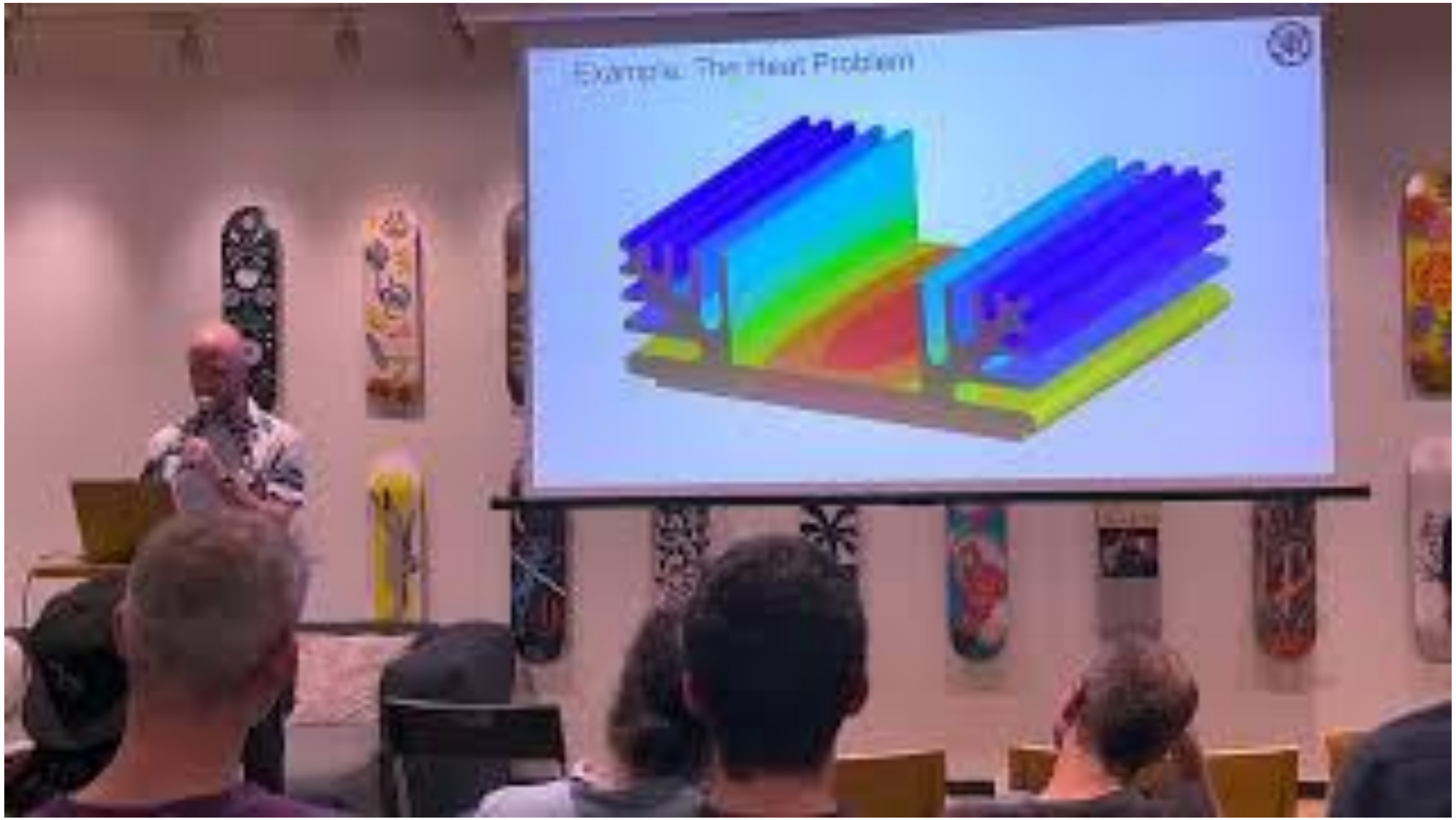
Special Thanks to

- Brian Schwind, who taught me about the `cxx` bridge and whose `opencascade-rs` crate served as both inspiration and copyable setup.
- Sjors Donkers, who was my source of wisdom for writing modern C++.
- Luke Peterson, who helped give feedback on this presentation.

Thanks for Listening!

https://mkovaxx.net

Rust-oriented talk with more details:



Appendix

MFEM-rs: Project Structure

Three Rust crates (packages) in a Cargo workspace:

- `mfem`
  - the idiomatic Rust API
  - traits equivalent to C++ base classes
- `mfem-sys`
  - bind to MFEM using the `cxx` crate
  - enforce ownership rules
- `mfem-cpp`
  - hook into on-system MFEM or build from source