Implementing your SasModels in Futhark

It goes much faster

Mikkel Storgaard Knudsen

Injecting Futhark into SasModels

Basic implementation strategy

- Adding Futhark support to SasModels is very non-invasive.

Basic implementation strategy

- Adding Futhark support to SasModels is very non-invasive.
- It is largely a question of adding a third kernel type (FutKernel) to exist along the two other available types, PyKernel and GpuKernel. SasModels itself behaves the same way no matter the kernel type used; as long as the kernel can be initialized and called by SasModels.
  Therefore we can implement FutKernel so it is initialized (and given q-values and other details) by SasModels.

Basic implementation strategy

- Adding Futhark support to SasModels is very non-invasive.
- It is largely a question of adding a third kernel type (FutKernel) to exist along the two other available types, PyKernel and GpuKernel. SasModels itself behaves the same way no matter the kernel type used; as long as the kernel can be initialized and called by SasModels.
  Therefore we can implement FutKernel so it is initialized (and given q-values and other details) by SasModels.
- When running the computations themselves, the calculations are performed within a dynamically loaded, precompiled, highly optimized Futhark kernel, and the results are returned to SasModels.

Basic implementation strategy

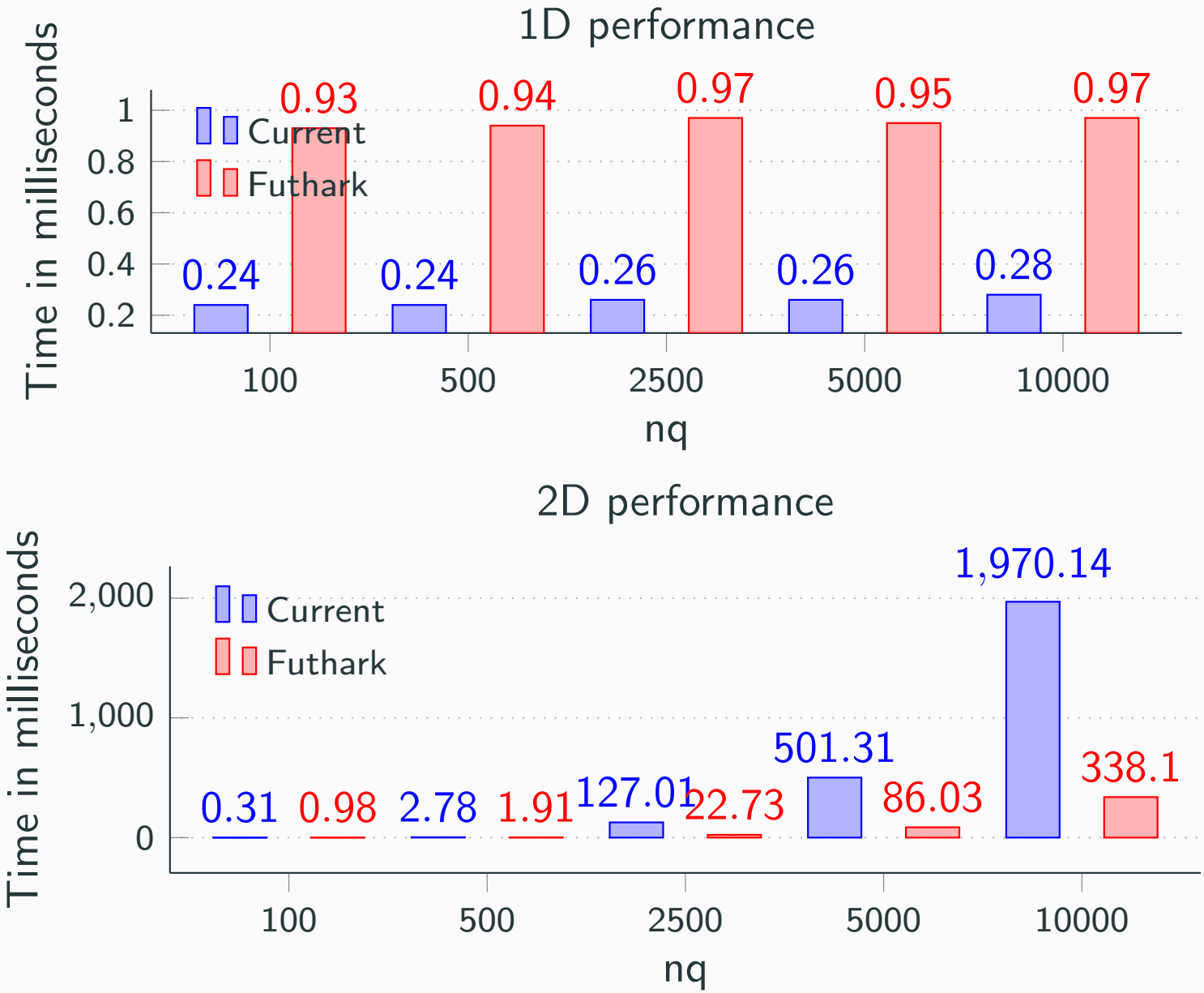
7 sasmodels/core.py — View

```
@@ -21,6 +21,7 @@
 from . import mixture
 from . import kernelpy
 from . import kerneldll
+from . import kernelfut
 from . import custom

 if os.environ.get("SAS_OPENCL", "").lower() == "none":
@@ -229,7 +230,11 @@ def build_model(model_info, dtype=None, platform="ocl"):
     numpy_dtype, fast, platform = parse_dtype(model_info, dtype, platform)

     source = generate.make_source(model_info)
-    if platform == "dll":
+
+    if type(model_info.Iq) is dict and model_info.Iq["model_is_futhark"]:
+        return kernelfut.FutModel(model_info, numpy_dtype)
+
+    elif platform == "dll":
         #print("building dll", numpy_dtype)
         return kerneldll.load_dll(source['dll'], model_info, numpy_dtype)
     else:
```

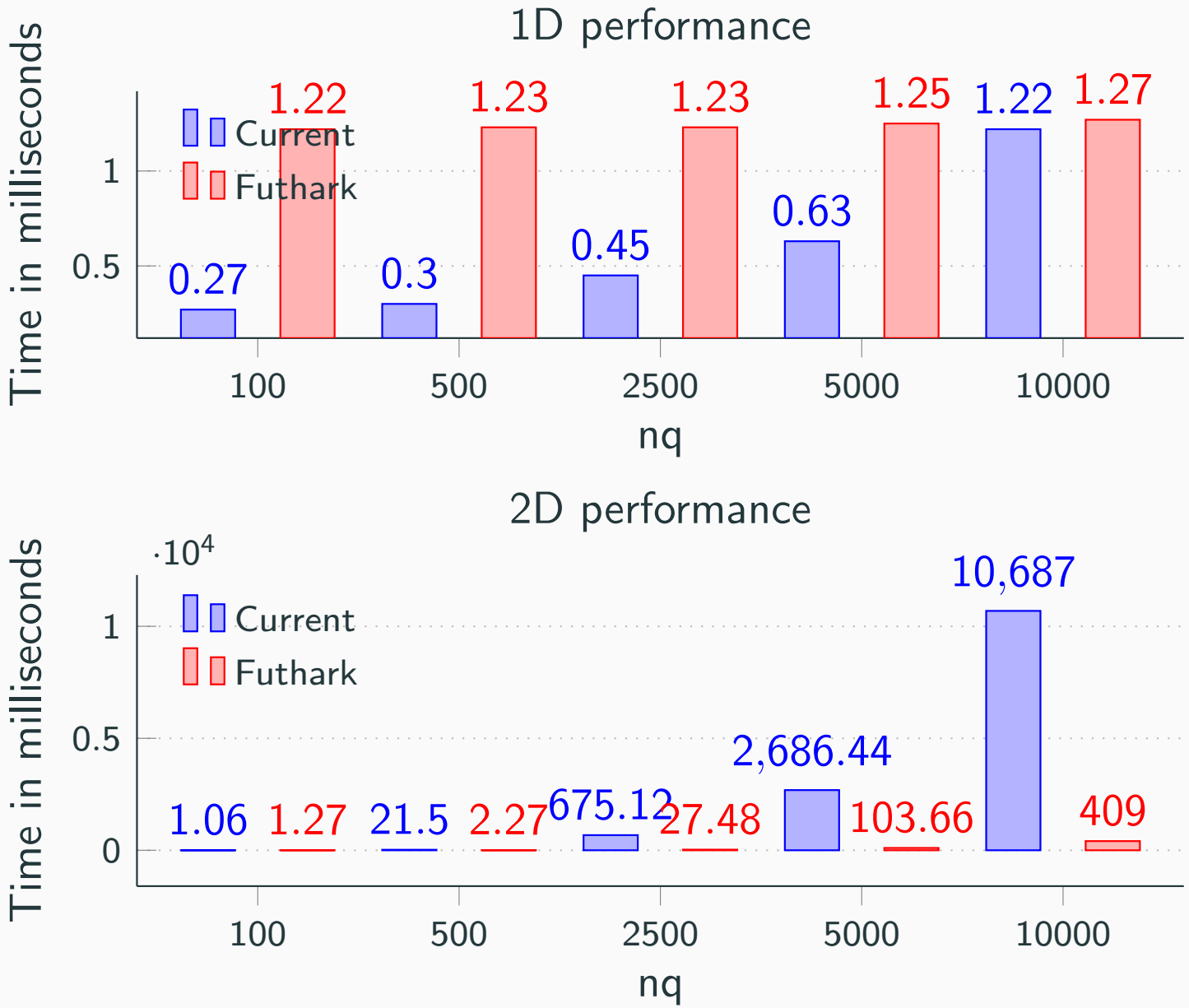
Figure 1: The complete contribution to already existing code

Performance

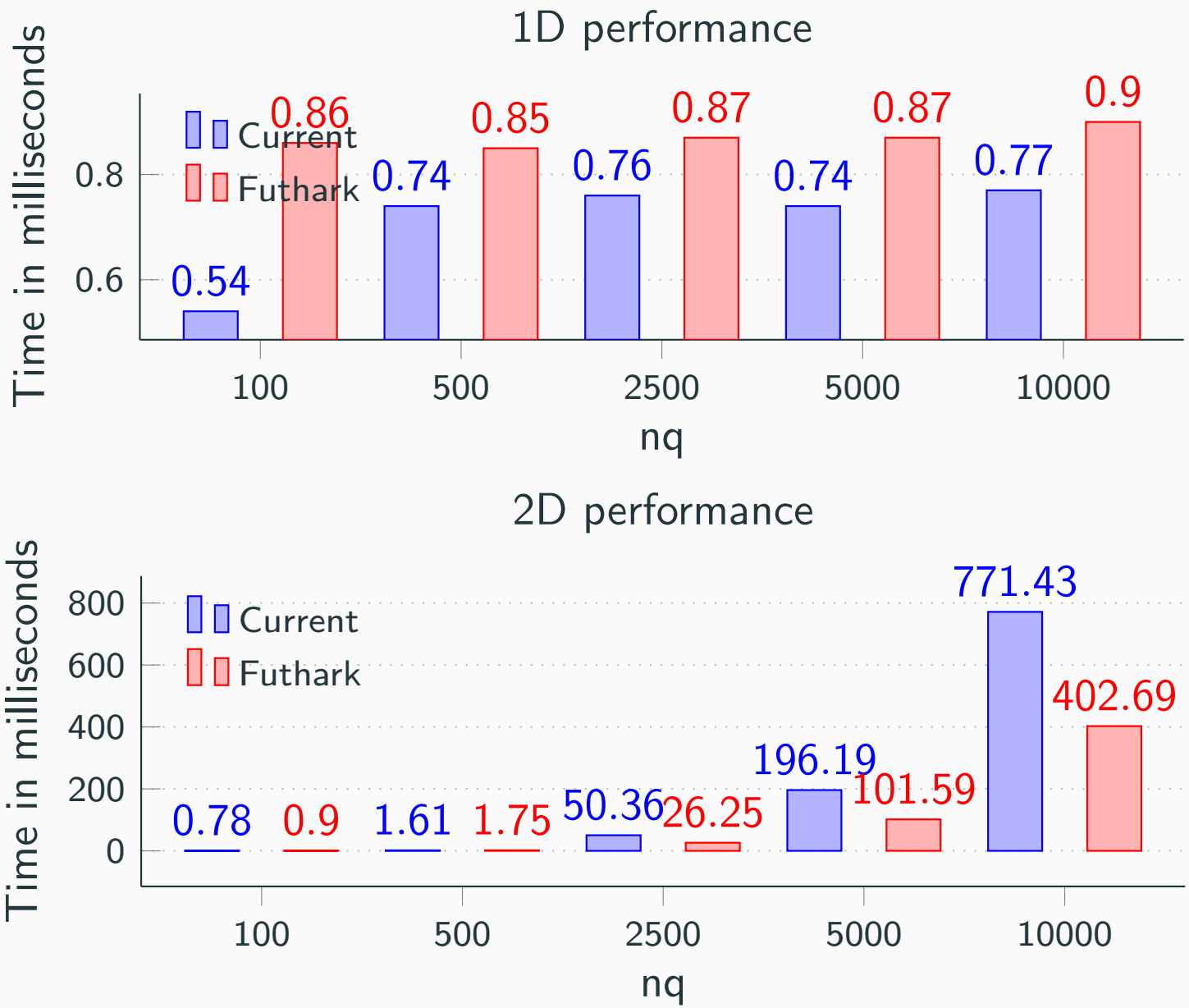
Average eval time for Line (trivial Python model)



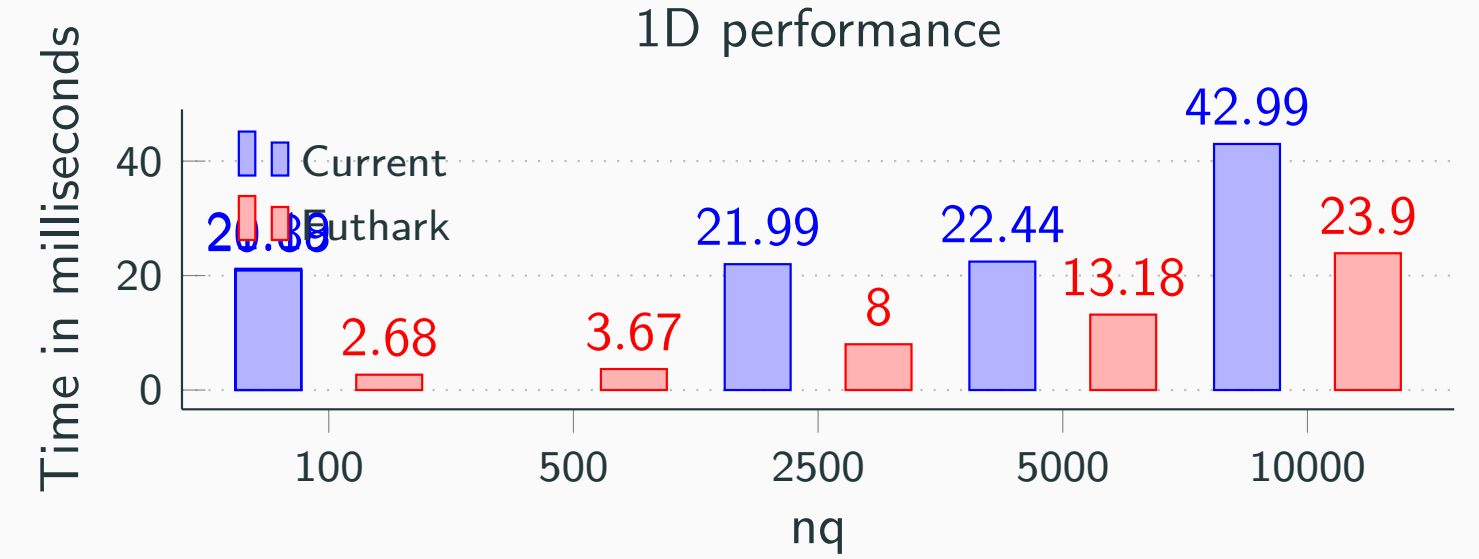
Average eval time for Broad Peak (Python model)

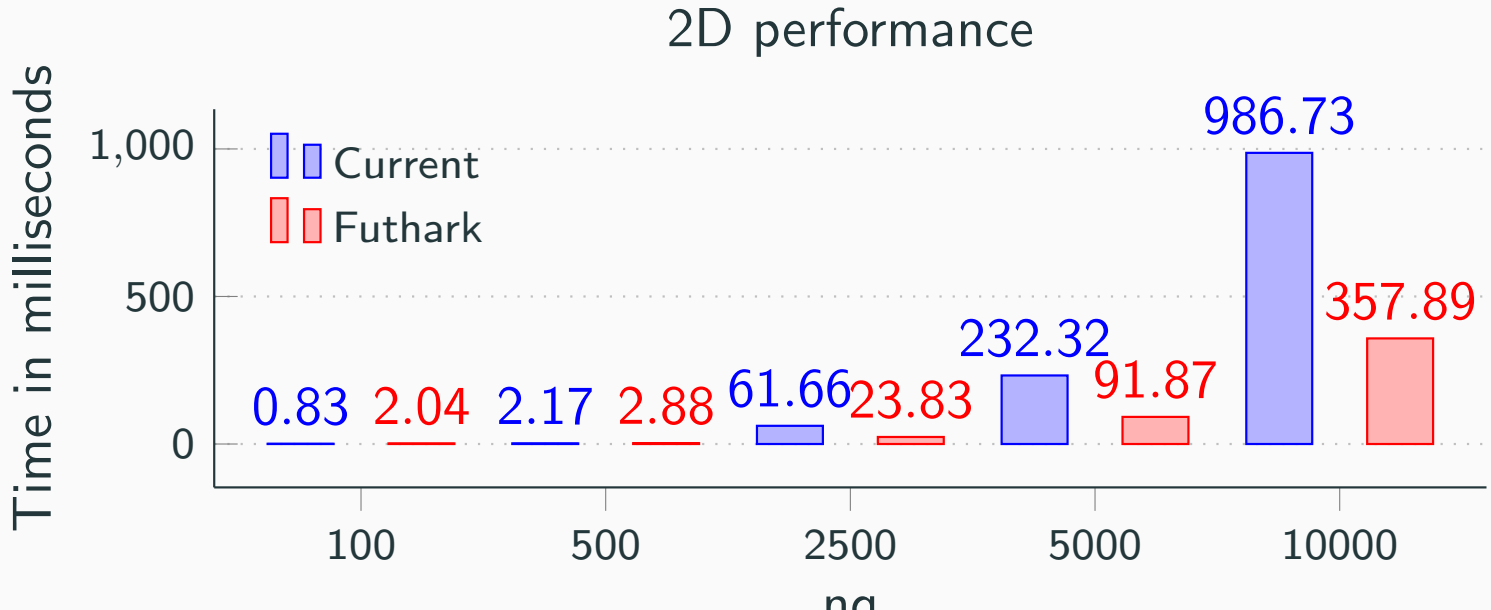


Average eval time for DAB (trivial OpenCL model)



Average eval time for core shell parallelepiped (OpenCL model)





Future work

Future work

SasModels in Futhark is not feature complete yet. We still need to implement:

- Polydispersion
- Magnetism
- Mixture models
- Futhark-side refactorings
- (various quality-of-life stuff)

We expect further experiments with more complicated models to show even larger performance boosts when using Futhark.