Computational Intelligence Methods for Solving Partial Differential Equations

An Experimental Study on Performance and Accuracy

Problem Description

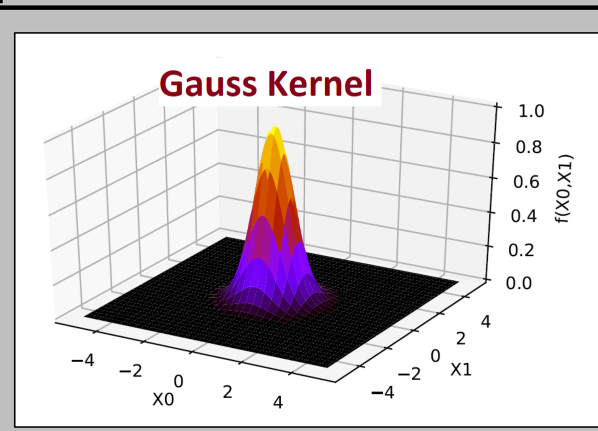
The solution to a PDE

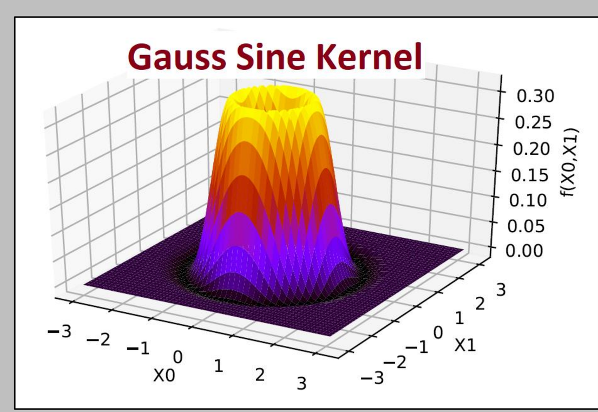
$$\boldsymbol{L}u(x) = f(x)$$
$$subjected\ to{:}\ \boldsymbol{B}u(x) = g(x)$$

is approximated by
a finite sum a Radial Basis Functions.
The best combination of these kernels is searched by minimising the Residual

$$R(x) = \boldsymbol{L}u_{apx}(x) - f(x)$$

of the PDE. Therefore, the fitness function is minimised by the heuristic optimisation algorithm JADE. Two different kernel-types are tested.





$$\mathrm{F}(u_{apx}(x)) = \frac{\sum_{i=1}^{nc} \xi(xi)\left[\boldsymbol{L}u_{apx}(xi) - f(xi)\right]^2 + \Phi \sum_{j=1}^{nB}\left[\boldsymbol{B}u_{apx}(xj) - g(xj)\right]^2}{nc+nb}$$

Algorithms

Differential Evolution is one of the most competitive heuristic optimisation algorithms. JADE is a modern variant of differential evolution that includes a self-adaption of the internal parameters. At the heart of JADE stands a population, where each individual represents a possible solution to the optimisation problem at hand.
After JADE terminates, the best individual is refined by a fast-converging deterministic local optimisation algorithm called Downhill Simplex. The coupling of two algorithms in such a way is called memetic algorithm.

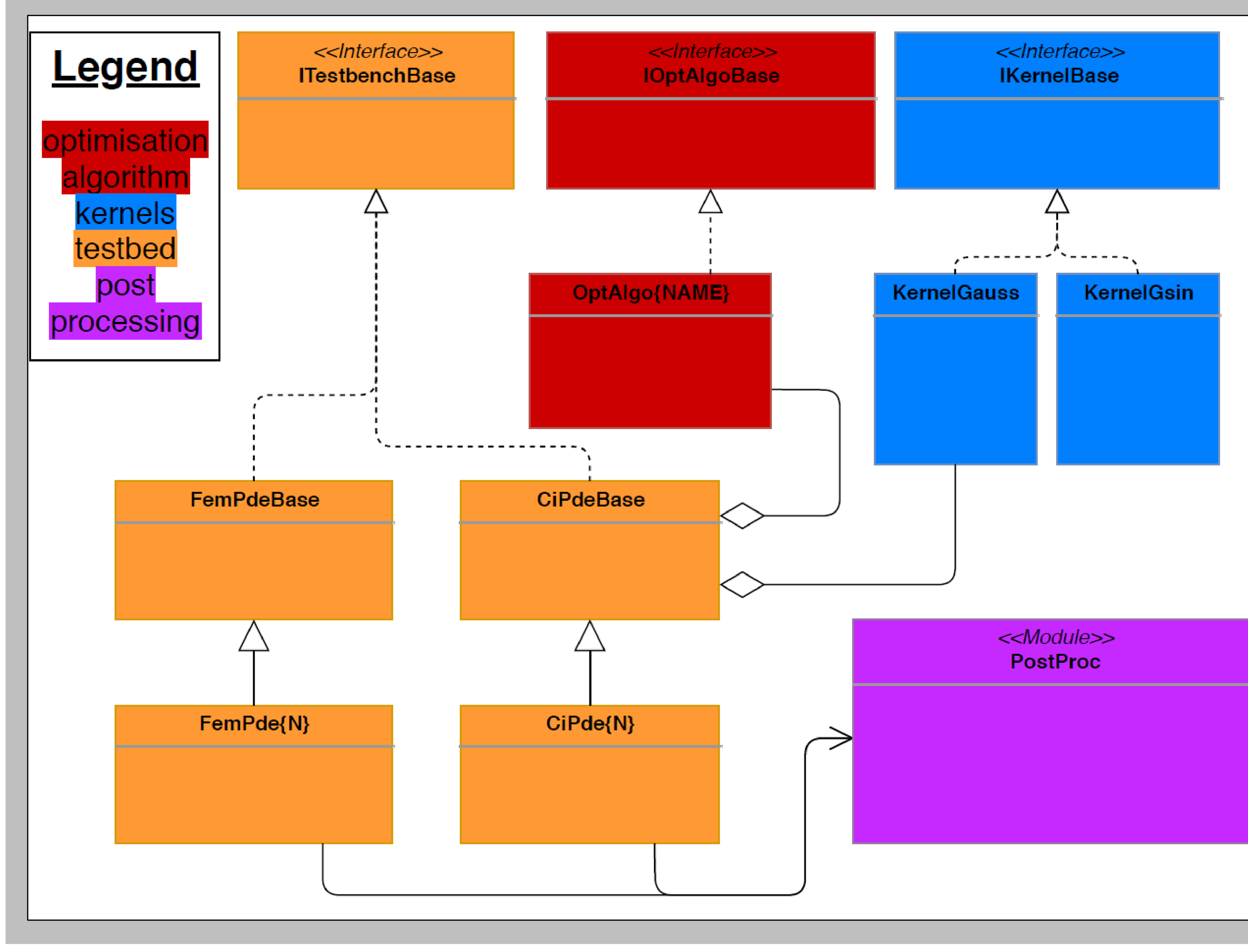
Altogether, four different algorithms are tested.

- serial memetic JADE with Gauss Kernels
- parallel memetic JADE with Gauss Kernels
- parallel memetic JADE with adaptive number of Gauss Kernels
- parallel memetic JADE with Gauss Sine Kernels

Experimental Design

The solver performance is evaluated in solving-time, memory usage and achieved approximation quality.
An extensive testbed of 11 Poisson equations is formed. To every PDE, the corresponding analytical solution in known.
Currently, the go-to approach for solving elliptic PDEs is an FEM solver. Therefore, the state of the art open source FEM solver NGSolve is used.
To account for the statistical influence, all experiments are redone for 20 independent replications. Replicability is ensured by calculating the statistical significance at α=0.05.

To set up experiments fast and in a standardised way, a comprehensive software structure is designed and implemented in Python.
The UML Class Diagram shows the key components and their connections. Some post-processing methods for saving and loading experiment objects as well as generating plots and calculating statistical indicators are summarised in the post-processing module.



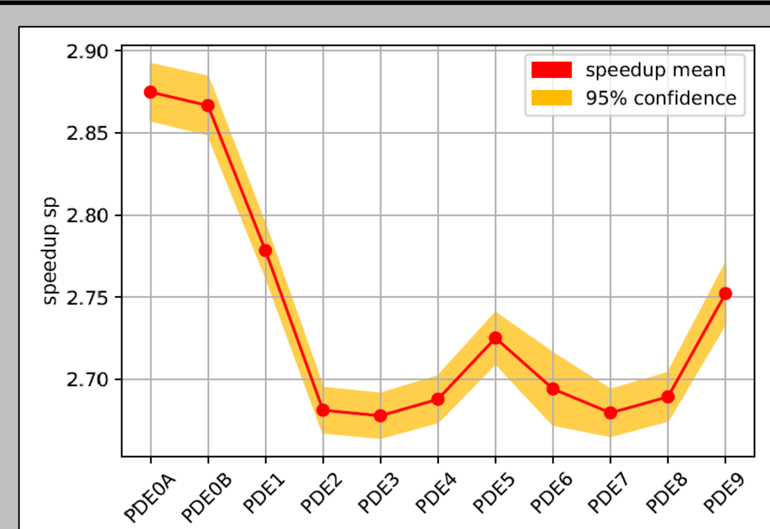
Results

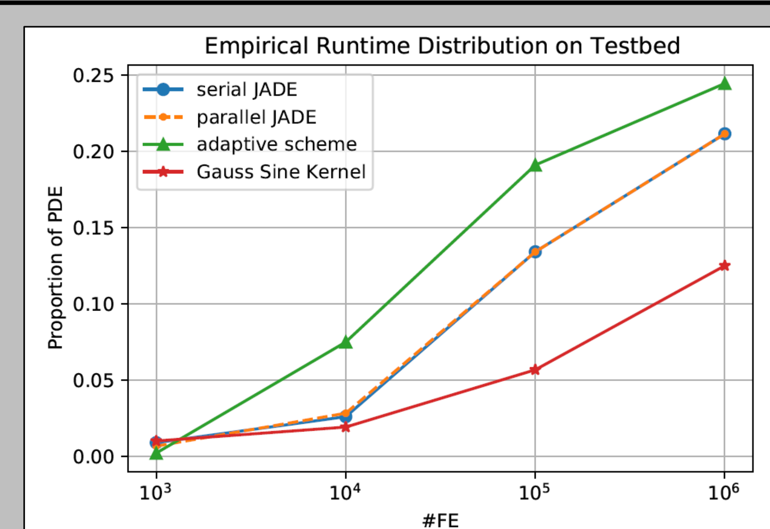
Generally, the usage of JADE can not replicate the results by similar algorithms from the literature.

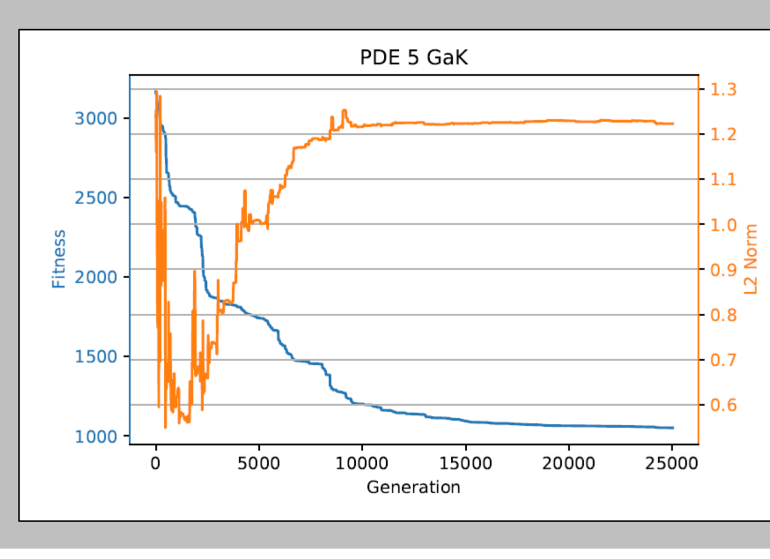
Comparing the solving time of the serial and the parallel algorithm shows a significant speedup. The speedup ranks between 2.6 and 2.9, depending on the testbed problem. Fitness functions that need more operations and thus take longer to evaluate, experience a better speedup.

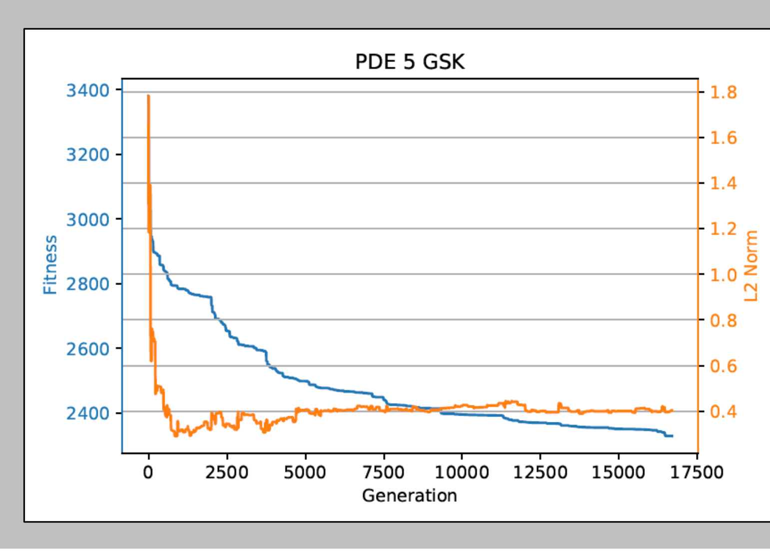
The quality of a solution is defined by its distance (L2) to the analytical solution.
On PDE 5 it can be observed that while the fitness value decreases with more generations, the quality does not necessarily get better.
This behaviour can be counteracted by using a different type of Kernel, in this specific case of PDE 5 a Gauss Sine Kernel is successful.

The Empirical-Runtime-Distribution shows the percentage of testbed functions, that can be successful solved after various number of function evaluations. Successful means, that the L2 norm is smaller than a specific target value.









Netgen/NGSolve

https://www.anaconda.com/

ANACONDA

https://www.anaconda.com/

Master
Mechatronics
2020

Studierende/r
Nicolai Schwartze, BSc
https://github.com/nicolai-schwartze/masterthesis

Betreuende Lehrperson der FHV
Dr.-Ing. Steffen Finck

FH Vorarlberg
University of Applied Sciences