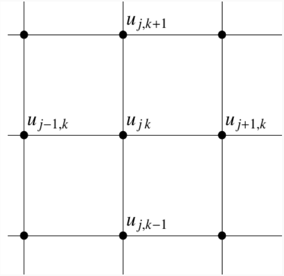
Partition-of-Unity Based Error Indicator for Local Collocation Meshless Methods

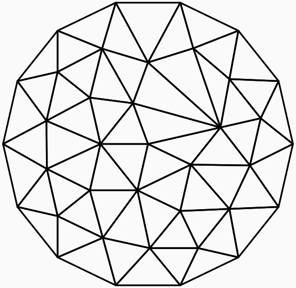
Jure Slak

"Jožef Stefan" Institute, Parallel and Distributed Systems Laboratory

1. 10. 2021, Mipro 2021

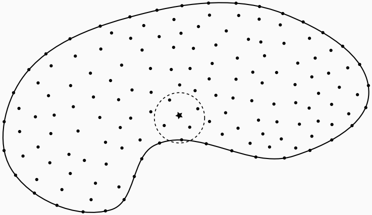
- Classical approaches:
  Finite Difference Method, Finite Element Method





- Problems: inflexible geometry, mesh generation
- Response: mesh-free methods (EFG, MLPG, FPM)

Domain discretization:



- Points x_i on the boundary and in the interior
- Point neighborhoods $N(x_i)$

Classical Finite Differences:

$$u''(x_i) \approx \frac{1}{h^2}u(x_{i-1}) - \frac{2}{h^2}u(x_i) + \frac{1}{h^2}u(x_{i+1})$$

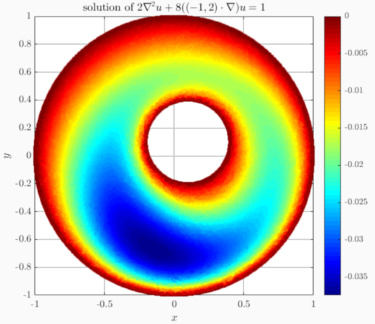
Generalized Finite Differences:

$$(\mathcal{L}u)(x_i) \approx \sum_{x_j \in N(x_i)} w_j^i u(x_j)$$

+ exactness for a certain set of functions (e.g. monomials)

Problem:

$$\mathcal{L}u = f \quad \text{on } \Omega,$$
$$u = u_0 \quad \text{on } \partial\Omega,$$



1. Discretize domain Ω
2. Find neighborhoods $N(x_i)$
3. Compute weights $\boldsymbol{w}^i$ for approximation of $\mathcal{L}$ over $N(x_i)$
4. Assemble weights in a sparse system $Wu = f$
5. Solve the sparse system $Wu = f$
6. Approximate/interpolate the solution

Adaptivity

The solution might not be good enough.

If we knew which regions had the highest error, we could increase the density of nodes there $\rightarrow$ h-adaptivity

Giving an estimate of the error based on the domain and the computed solution is a job of an **error indicator**.

Adaptivity

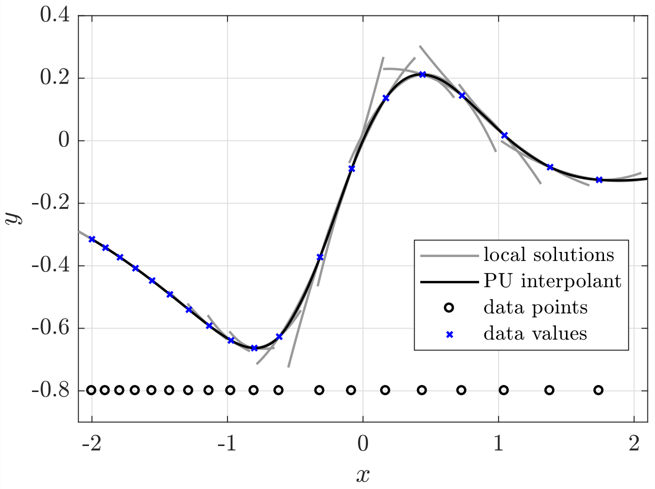
Algorithm 1 Adaptive solution algorithm.

Input: Initial discretization $D^{(0)} = (X^{(0)}, \tau^{(0)}, S^{(0)}, \vec{n}^{(0)})$ of Ω.
Input: Global error tolerance ϵ.
Input: Maximal number of adaptive iterations J.
Output: The adaptive solution u.

function $\textsc{adaptiveSolve}(\Omega,\ D^{(0)},\ \epsilon,\ J)$
    for j $\leftarrow$ 0 **to** J **do**
        $u^{(j)} \leftarrow \textsc{solve}(D^{(j)})$
        if $j < J$ **then**
            $e^{(j)} \leftarrow \textsc{estimateError}(D^{(j)}, u^{(j)})$
            $e_m^{(j)} \leftarrow \textsc{mean}(e^{(j)})$
            if $e_m^{(j)} < \epsilon$ **then**
                return $u^{(j)}$
            end if
            $R^{(j)} \leftarrow \{i \mid i = 0, \dots |X^{(j)}| - 1;\ e^{(j)}[i] > e_m^{(j)}\}$ ▷ Above-average error.
            $D^{(j+1)} \leftarrow \textsc{refine}(\Omega, D^{(j)}, R^{(j)})$
        end if
    end for
    return $u^{(J)}$ ▷ Maximal number of iterations reached.
end function

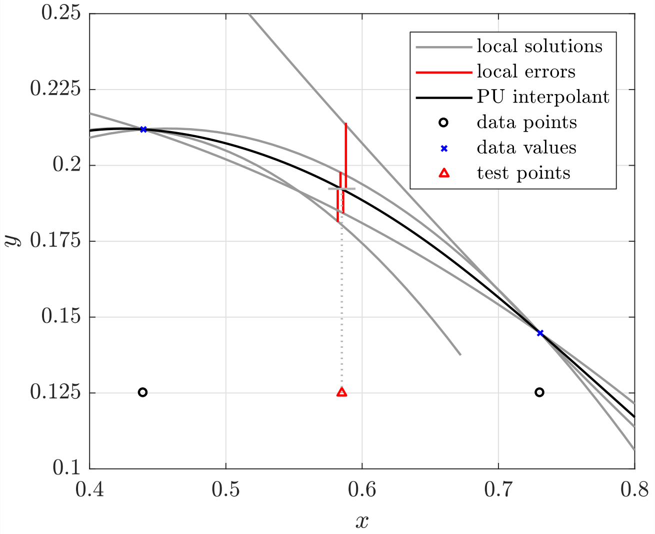
Partition-of-unity error indicator

The global solution is made up of glued local solutions.



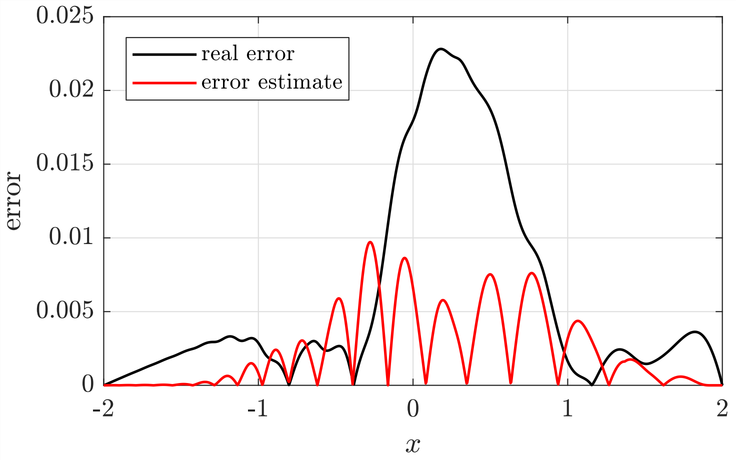
Partition-of-unity error indicator – zoom

We compute the variability of local solutions in test points.



Partition-of-unity error indicator – sample

Comparison with the real error for our example.



Automatic adaptivity – Poisson example

Consider a problem in d-dimensions (for $d = 2$ and $d = 3$)

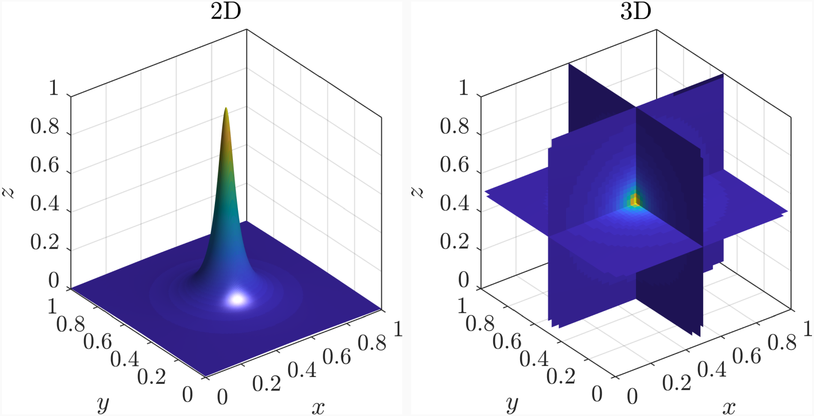
$$\begin{aligned} \nabla^2 u &= f \quad \text{in} \quad \Omega, \\ u &= g \quad \text{on} \quad \partial\Omega, \end{aligned} \tag{1}$$

where $\Omega = [0, 1]^d$ and the f and g are computed from the manufactured solution

$$u(\boldsymbol{x}) = \frac{1}{25\|4\boldsymbol{x} - 2\|^2 + 1}.$$

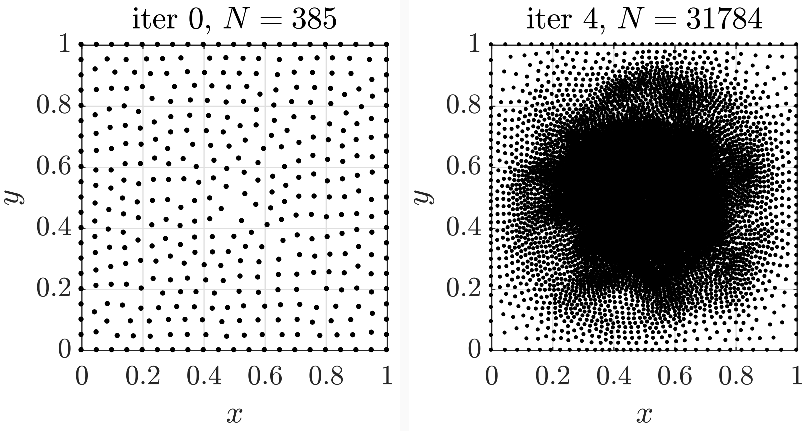
Automatic adaptivity – Poisson example

Solution:

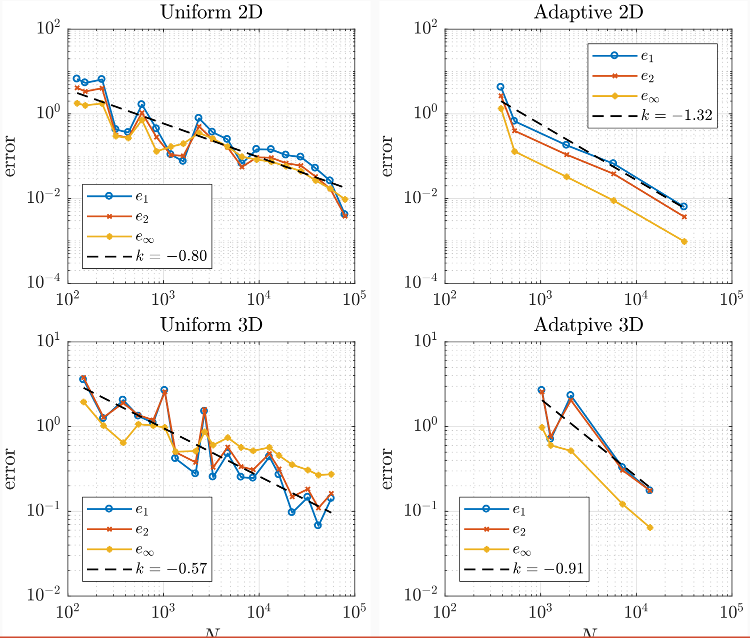


Automatic adaptivity – Poisson example

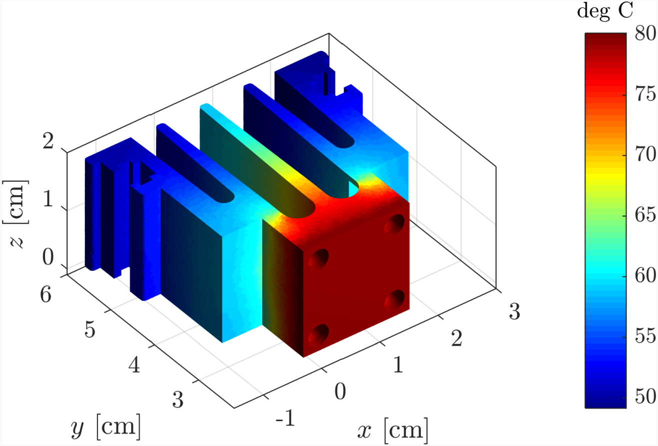
Distribution of nodes:



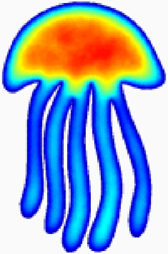
Automatic adaptivity – results



3D Heatsink model



All computations were done using open source Medusa library.



Medusa

Coordinate Free Mehless Method implementation

http://e6.ijs.si/medusa/

Thank you for your attention!

Acknowledgments: ARRS research core funding No. P2-0095.