

If you lack time or ability to indulge into parallel programming, this toolbox is what you need.

Without efforts you can convert your Diffpack simulator into a powerful parallel code, running with optimized speed-up.



**inuTech GmbH
Fürther Str. 212
90429 Nürnberg
Germany**

**Phn: +49 911 32 38 43 - 0
Fax: +49 911 32 38 43 - 43
Eml: diffpack@inutech.de**

**www.diffpack.com
www.inutech.de**

Easy and Affordable Parallel Computing

Traditionally, parallel computing has been a field limited to an elite of dedicated experts. Not only have the required computers been prohibitively expensive, but the practical programming has been complicated and time consuming as well.

With the development of high-speed networking and low-cost machine clusters, parallel equipment has become considerably more affordable.

For Diffpack users, parallel computing has actually become easy! All you need is to add a few statements to your program, link it to the Parallel Toolbox and it will run on a multi-processor architecture with optimized speed-up.

Linear Algebra Parallelization for PDEs

The Parallel Toolbox I is designed to parallelize the linear algebra level of your Diffpack based PDE applications. Using a data-parallel approach, it distributes your data structures and calculations automatically to a set of processors. Data partitioning is performed in accordance with the topology of the computational mesh, which ensures balanced communication and high speed-up.

The following example shows the result of parallelizing a finite element simulator for a problem of the form $-\nabla(K(x)\nabla u) = f(x)$ on a highly unstructured mesh.

# Processors	CPU Seconds	Speed-up
1	420	N/A
3	200	2.10
4	156	2.69
6	84	5.01
8	60	6.97
12	38	10.99
16	28	14.83

The problem has about 130,000 degrees of freedom, overlapping sub-meshes and uses BiCGStab with block ILU preconditioning as linear solver.

Equally Flexible as in the Sequential Setting

Basically, the parallelization works by splitting the full mesh into an array of overlapping sub-meshes. Each sub-mesh defines its local contribution to the total algebraic system, while common mesh nodes define the dependency between the local contributions. Each sub-mesh is assigned to one process. The processes run in parallel and data is communicated as needed in the event of sub-mesh overlaps.

When setting up your application to run in parallel, you can select most Diffpack run-time options that are available in the sequential setting. In particular you can use structured or unstructured meshes in 1, 2 and 3 space dimensions and you can select different iterative solvers, preconditioners, elements, etc.

(turn page)



Easy to Configure

The run-time options you can use for the parallelization setup basically include the distribution of processes to processors (often one-to-one), algorithms for mesh partitioning and the size of sub-mesh overlaps. You can select to read each sub-mesh from file or partition a global unstructured mesh with METIS or your own mesh partition algorithm. Structured meshes can be constructed in parallel.

This example shows the standard Poisson equation ($-\nabla^2 u = f$) on the unit square using a structured mesh generated in parallel:

# Processors	CPU Seconds	Speed-up
1	1025	N/A
2	493	2.08
4	254	4.03
6	172	5.94
8	124	8.25
12	79	12.95
16	58	17.68

The problem has about 230,000 degrees of freedom, overlapping sub-meshes and uses Conjugate Gradients without preconditioning as the linear solver. The super-linear speed-up is due to cache effects.

Applicable from PCs in Network to Dedicated Parallel Machines

The Parallel Toolbox I uses MPI as the communication protocol. As long as MPI is installed on your system, you can parallelize Diffpack applications on dedicated parallel hardware as well as processors connected in a network.

The toolbox is licensed in terms of number of processes rather than the number of users, which is the case for other Diffpack units. This means that you can allocate all processes to one running Diffpack application or share the available processes between several Diffpack applications.