Adapting a multibody system simulator to auto-tuning linear algebra routines

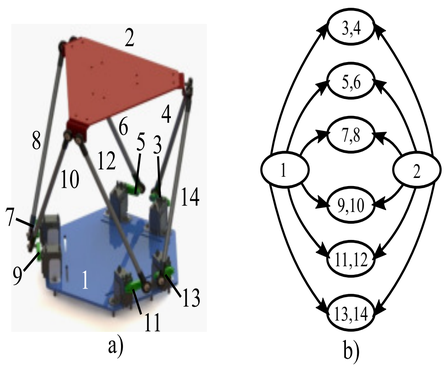
Jesús Cámara*, José-Carlos Cano•, Javier Cuenca*, Domingo Giménez•, Mariano Saura-Sánchez†

* Department of Engineering and Technology of Computers, University of Murcia, Spain
• Department of Computing and Systems, University of Murcia, Spain
† Department of Mechanical Engineering, Technical University of Cartagena, Spain

Multibody Systems (MBS)

A set of bodies interconnected through mechanical joints which allow combined movements among them.
Computational kinematics studies the movement of MBS from different approaches.
Modular approach (**Group Equations**): divide a MBS into a set of modules whose kinematics can be solved in a hierarchical order ⇒ facilitates parallelism.

Example: Stewart Platform



For each group one system of equations.
Six simultaneous systems.

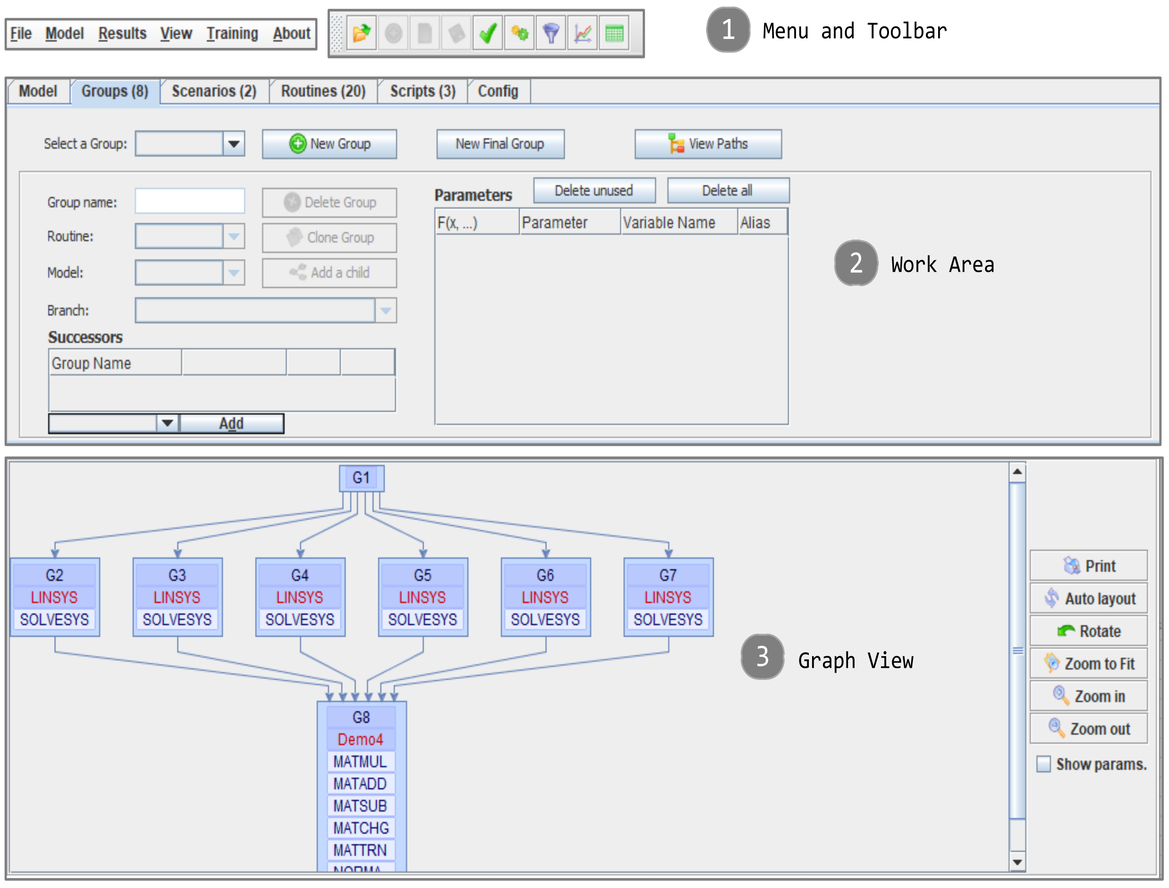
Stewart Platform: Computation

Algorithm 1 Schema of the Group Equations method for the Stewart Platform

```
for number of external iterations do
    Solve kinematics of terminal
    for all structural components do
        //explicit parallelism, OpenMP
        for number of internal iterations do
            Solve kinematics of structural component
            //system of equations, basic libraries
        end for
    end for
end for
```

Several simultaneous equations.
Sparse matrices.

Parallel Simulator for Multi-Body Systems Based on Group Equations



Desktop application with three parts:

- `Menu and Toolbar`: access to the functionalities of the simulator for managing models.
  Models can be simulated (`Training`): generation of a database with information of execution time for the possible combinations of the nodes of the directed graph and their assignation to the computing units.
- `Work Area`: to manage a particular model.
  Groups of functions are created (nodes) and their connections are established.
  Each group contains a set of basic routines directly implemented or from basic libraries (MKL, M27, cuBLAS...)
  The scenarios represent the input for the simulation: matrix sizes and sparsity degree.
  A set of algorithmic parameters (OpenMP threads, number of GPUs...) whose values for low execution time are determined in the simulations.
- `Graph View`: shows the graph of the model (in the example, Stewart platform).

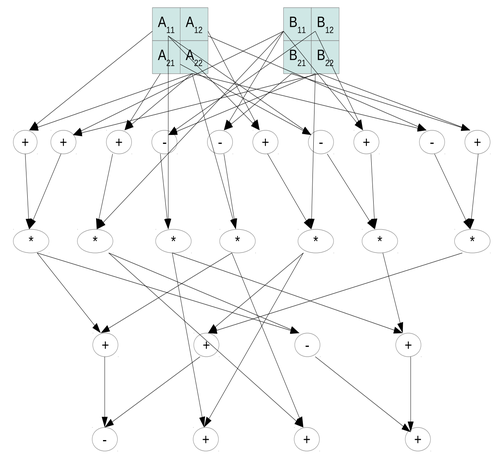
The basic functions are trained ***off-line*** for the given scenarios.
At running time the decision of the parameters with which the simulation is carried out is taken ***on-line***.

José-Carlos Cano, Javier Cuenca, Domingo Giménez, Mariano Saura-Sánchez, Pablo Segado-Cabezos: A parallel simulator for multibody systems based on group equations. The Journal of Supercomputing 75(3): 1368-1381 (2019)

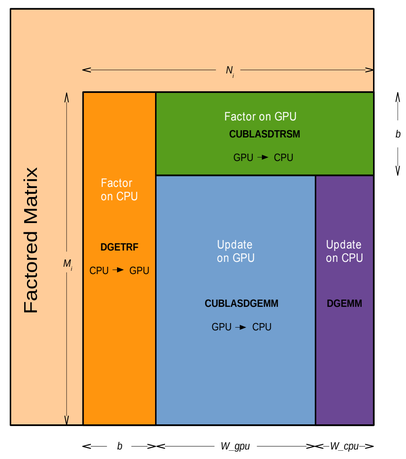
Adaptation for auto-tuning linear algebra routines

Linear algebra routines decompose in computations with smaller matrices.

Strassen multiplication

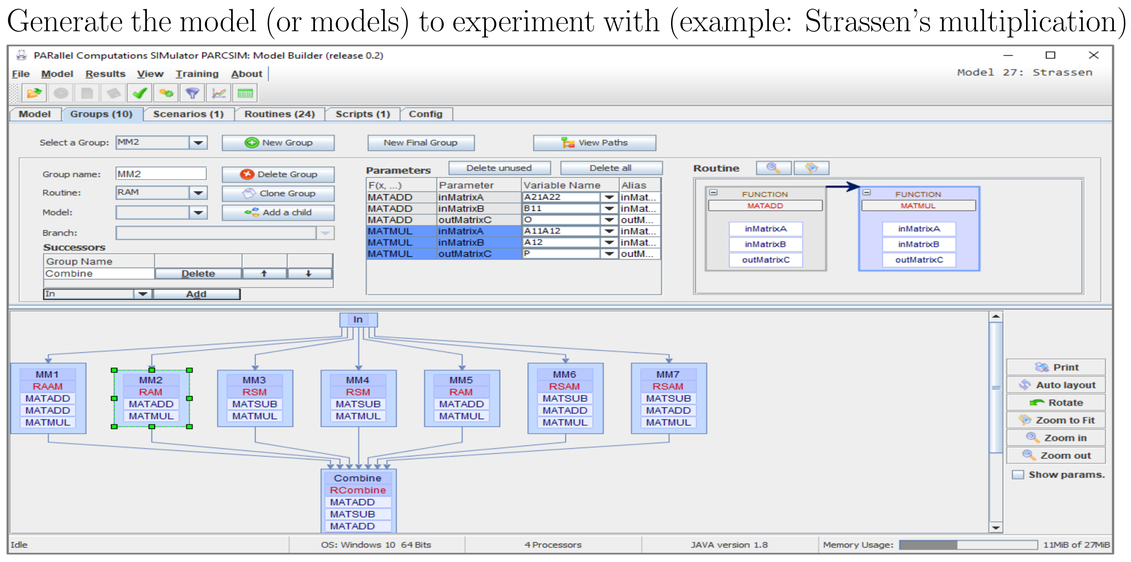


LU factorization

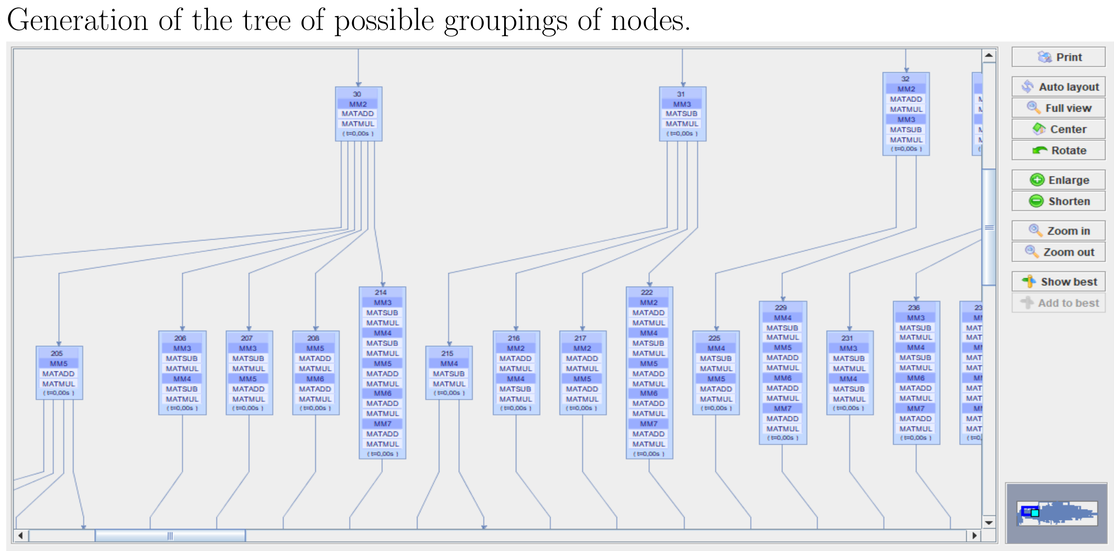


Application of the simulator to LAR auto-tuning

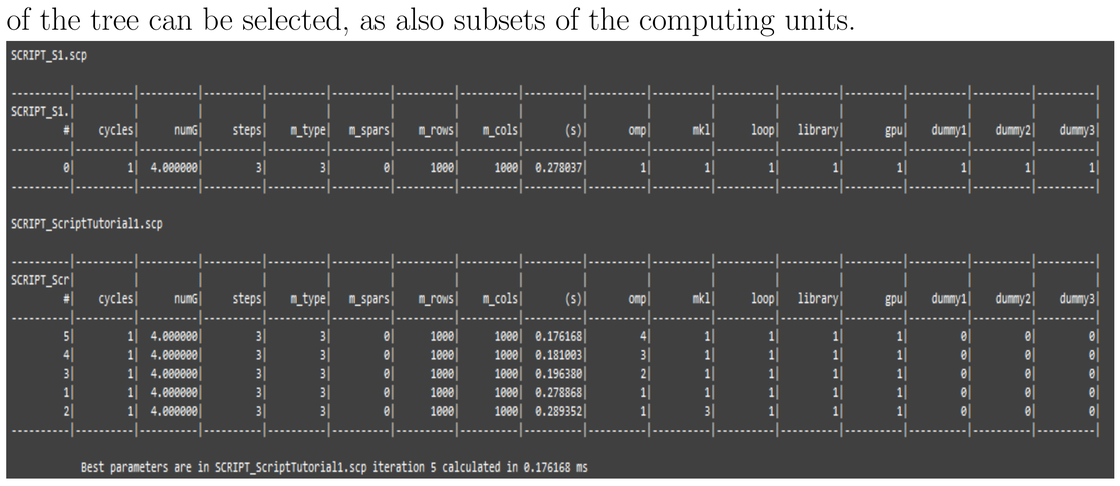
- Generate the model (or models) to experiment with (example: Strassen's multiplication).



- Generation of the tree of possible groupings of nodes.



- Simulation and storage of the results for posterior use for other problem sizes. Branches of the tree can be selected, as also subsets of the computing units.



```
SCRIPT_S1.scp

SCRIPT_S1.|          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
        #|   cycles|      numG|     steps|   m_type|   m_spars|   m_rows|   m_cols|      (s)|      omp|      mkl|     loop|  library|      gpu|   dummy1|   dummy2|   dummy3|
         |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
        0|        1| 4.000000|        3|        3|        0|     1000|     1000| 0.278037|        1|        1|        1|        1|        1|        1|        1|        1|

SCRIPT_ScriptTutorial1.scp

SCRIPT_Scr|          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
        #|   cycles|      numG|     steps|   m_type|   m_spars|   m_rows|   m_cols|      (s)|      omp|      mkl|     loop|  library|      gpu|   dummy1|   dummy2|   dummy3|
         |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
        5|        1| 4.000000|        3|        3|        0|     1000|     1000| 0.176168|        4|        1|        1|        1|        1|        0|        0|        0|
        4|        1| 4.000000|        3|        3|        0|     1000|     1000| 0.181003|        3|        1|        1|        1|        1|        0|        0|        0|
        3|        1| 4.000000|        3|        3|        0|     1000|     1000| 0.196380|        2|        1|        1|        1|        1|        0|        0|        0|
        1|        1| 4.000000|        3|        3|        0|     1000|     1000| 0.278868|        1|        1|        1|        1|        1|        0|        0|        0|
        2|        1| 4.000000|        3|        3|        0|     1000|     1000| 0.289352|        1|        3|        1|        1|        1|        0|        0|        0|

Best parameters are in SCRIPT_ScriptTutorial1.scp iteration 5 calculated in 0.176168 ms
```

Further improvements

- Facilitate the generations of alternative models for a problem, and the selection of the best model.
- Reuse of the information generated for a routine or model in other routines which use them.
- Integration of the MBS in an auto-tuning system to generate auto-tuned libraries.