Massive Parallelization of Multibody System Simulation

Michael Valášek, Ladislav Mráz

Department of Mechanics, Biomechanics and Mechatronics, Faculty of Mechanical Engineering, Czech Technical University in Prague Technická 4, 166 07 Praha 6, Czech Republic

Corresponding author: Michael.Valasek@fs.cvut.cz

Abstract

This paper deals with the decrease in CPU time necessary for simulating multibody systems by massive parallelization. The direct dynamics of multibody systems has to be solved by a system of linear algebraic equations. This is a bottleneck for the efficient usage of multiple processors. Simultaneous solution of this task means that the excitation is immediately spread into all components of the multibody system. The bottleneck can be avoided by introducing additional dynamics, and this leads to the possibility of massive parallelization. Two approaches are described. One is a heterogeneous multiscale method, and the other involves solving a system of linear algebraic equations by artificial dynamics.

Keywords: multibody system; dynamics; simulation; massive parallelization; heterogeneous multiscale method; additional dynamics.

1 Introduction

The CPU time required to carry out a simulation of dynamic systems is too largeto be solved on current processors. The process can be speeded up by parallelization. It is forecast that parallel computers will be widely used in the near future, and it is necessary to develop new formalisms for effective exploitation of the computing cores. Many techniques have been developed for computing on multiple processors [1–7], but only a few of them have been developed for massive parallelization [5–7]. This paper investigates the main problem preventing efficient usage of multiple processors. It brings together two recently developed approaches [8–9].

2 Problem of massive parallelization

The traditional solution of multibody dynamics always includes solving a system of linear algebraic equations (LAE). It is solved either by inversion of the mass matrix $\boldsymbol{M}$:

$$\boldsymbol{M}\ddot{\boldsymbol{q}} = \boldsymbol{F}, \tag{1}$$

where $\boldsymbol{q}$ are the generalized coordinates and $\boldsymbol{F}$ the forces, or by inversion of the overall matrix including the mass matrix $\boldsymbol{M}$ and the Jacobi matrix Φ of constraints

$$\begin{bmatrix} \boldsymbol{M} & \Phi^{\mathsf{T}} \\ \Phi & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{s}} \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{Q_1} \\ \boldsymbol{Q_2} \end{bmatrix}, \tag{2}$$

where $\boldsymbol{s}$ are the dependent coordinates, $\boldsymbol{Q_1}$ is the force component and $\boldsymbol{Q_2}$ the vector of the acceleration remainder resulting from the second time derivatives of the constraints.

If an excitation enters the system in the i-th force element F_i, then it is immediately spread among all coordinates $\boldsymbol{q}$ (or $\boldsymbol{s}$) through the solution of the mass matrix (or the system matrix). Physically, this means spreading the information inside the system by infinite velocity. In nature, however, its velocity is limited by the velocity of acoustic waves and in the limit by the velocity of light.

Massive parallelization of the solution of LAE can reduce the computational complexity to $\mathcal{O}(n)$ [7], but it never achieves $\mathcal{O}(1)$, i.e., independence from the size of the system.

A solution for this problem might therefore be to introduce finite velocity of spreading the information inside the system. This is achieved in the following two approaches by introducing flexibility into the kinematic constraints and by completely artificial introduction of dynamics into the solution of LAE.

3 Heterogeneous multiscale method

Classical approaches for solving multibody dynamics usually use rigid joints or constraints, leading to interconnected equations of motion (EOM). In order to obtain a set of disconnected EOM, the classical joints are replaced by flexible joints with high stiffness and appropriate damping, in fact providing a more realistic description of the joints. An example of such a replacement for a planar multibody system connected by revolute joints is shown in Fig. 1.

Using springs and dampers, it is possible to assemble a set of decoupled EOM for the kinematical chain
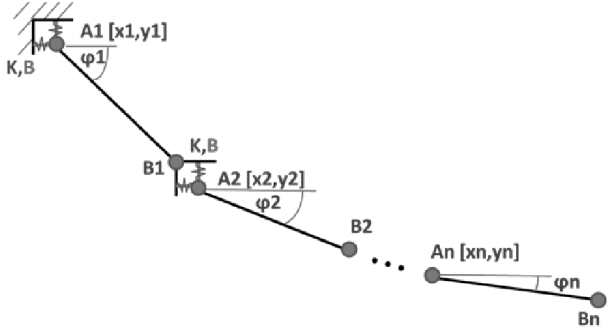
Figure 1: Model of a kinematical chain with joints replaced by springs and dampers.

in Fig. 1. Each body in a plane is described by two Cartesian coordinates of point A_i and by an angle measured from the global x axis, and the set of coordinates for describing the position of body i in the chain is $s_i = [x_i, y_i, \varphi_i]^\mathsf{T}$. Using Lagrange equations of the second type, the EOM for each body can be written in the following form

$$M_i s_i = F_i, \qquad i = 1, 2, \ldots, n, \qquad (3)$$

where M_i is the mass matrix of body i with the following structure:

$$M_i = \begin{bmatrix} m_i & 0 & -m_i p_i\, \mathrm{s}\varphi_n \\ 0 & m_i & m_i p_i\, \mathrm{c}\varphi_n \\ -m_i p_i\, \mathrm{s}\varphi_n & m_i p_i\, \mathrm{c}\varphi_n & I_{A_i} \end{bmatrix},$$

where $\mathrm{c}\varphi_i = \cos\varphi_i$, $\mathrm{s}\varphi_i = \sin\varphi_i$, p_i is the position of the center of mass (Fig. 2), m_i is the mass of body i and I_{A_i} is the moment of inertia of body i with respect to point A_i. If point A_i was the center of mass, the matrix M_i would be diagonal. The acceleration vector $\ddot{s}_i$ has the structure $\ddot{s}_i = [\ddot{x}_i, \ddot{y}_i, \ddot{\varphi}_i]^\mathsf{T}$. The generalized forces vector F_i contains the applied forces, the forces resulting from the presence of springs and dampers, and the Coriolis and centrifugal forces. It can be expressed as

$$F_i = \begin{bmatrix} -F_{i,x}^K - F_{i,x}^B + F_{i+1,x}^K \\ F_{i,y}^K + F_{i,y}^B - F_{i+1,y}^K \\ -(F_{i+1,x}^K + F_{i,x}^B)\ell_i\, \mathrm{c}\varphi_i \end{bmatrix}$$
$$+ \begin{bmatrix} F_{i+1,x}^B + F_{i,x}^A \\ -F_{i+1,y}^B + F_{i,y}^A \\ -(F_{i+1,y}^K + F_{i+1,y}^B)\ell_i\, \mathrm{s}\varphi_i \end{bmatrix}$$
$$+ \begin{bmatrix} p_i m_i \dot{\varphi}_i^2\, \mathrm{c}\varphi_i \\ -G_i + p_i m_i \dot{\varphi}_i^2\, \mathrm{s}\varphi_i \\ -G_i p_i\, \mathrm{c}\varphi_i + M_i^A \end{bmatrix}, \quad (4)$$

where $F_i^A = [F_{i,x}^A, F_{i,y}^A]^\mathsf{T}$ and M_i^A are the resulting applied force and torque acting on body i and decomposed in center of mass S_i, $F_{i,x}^K$ are the forces in the
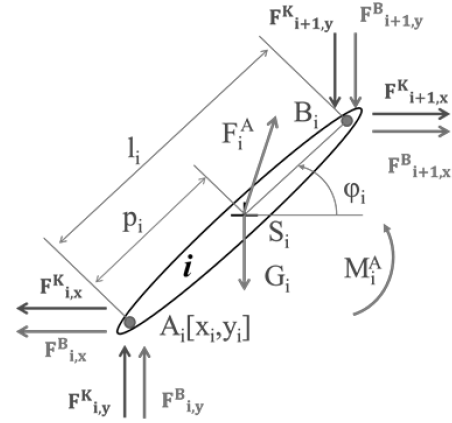

Figure 2: Single body in a kinematical chain.

springs, and $F_{i,x}^B$ are the forces in the dampers, which are expressed in the x axis direction between body i and $i - 1$. A similar notation holds for the y axis. The forces are depicted in Fig. 2 and are described by the equations

$$\begin{aligned} F_{i,x}^K &= K(x_i - x_{B_{i-1}}), \\ F_{i,y}^K &= K(-y_i + y_{B_{i-1}}), \\ F_{i,x}^B &= B(\dot{x}_i - \dot{x}_{B_{i-1}}), \\ F_{i,y}^B &= B(-\dot{y}_i + \dot{y}_{B_{i-1}}), \end{aligned} \qquad (5)$$

where K and B are constants of stiffness and damping, respectively. Point B_i is described by the coordinates x_{B_i} and y_{B_i}, and can be expressed as

$$\begin{aligned} x_{B_i} &= x_i + \ell_i\, \mathrm{c}\varphi_i, \\ y_{B_i} &= y_i + \ell_i\, \mathrm{s}\varphi_i, \end{aligned} \qquad (6)$$

and their time derivatives

$$\begin{aligned} x_{B_i} &= x_i + \ell_i \dot{\varphi}_i\, \mathrm{c}\varphi_i, \\ y_{B_i} &= y_i + \ell_i \dot{\varphi}_i\, \mathrm{s}\varphi_i, \end{aligned} \qquad (7)$$

Rewriting the EOM for the whole system using the equations of motion of single bodies obtained previously, the mass matrix (system matrix) has a block-diagonal structure (8), where each block corresponds to one body.

$$\begin{bmatrix} M_1 & & & \\ & M_2 & & \\ & & \ddots & \\ & & & M_n \end{bmatrix} \begin{bmatrix} \ddot{s}_1 \\ \ddot{s}_2 \\ \vdots \\ \ddot{s}_n \end{bmatrix} = \begin{bmatrix} F_1(t, S, \dot{S}) \\ F_2(t, S, \dot{S}) \\ \vdots \\ F_n(t, S, \dot{S}) \end{bmatrix}, \quad (8)$$
$$S = [s_1, s_2, \ldots, s_n]^\mathsf{T}, \quad \dot{S} = [\dot{s}_1, \dot{s}_2, \ldots, \dot{s}_n]^\mathsf{T}.$$

An analysis of (8), clearly shows that the EOMs are not interconnected and it is possible to compute the accelerations and to integrate them fully independently, and thus use parallel computing. The
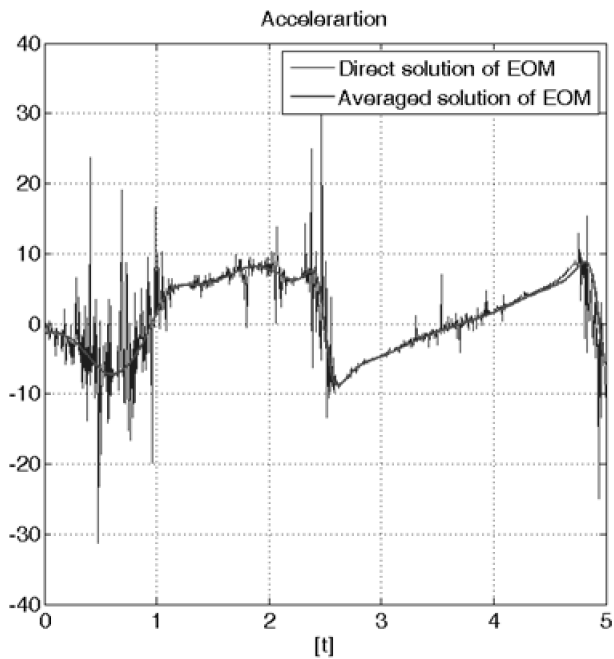
Figure 3: Comparison of the direct and averaged (HMM) solution at accelerations.

processors must only communicate with their neighbors for the spring and damper forces, and there has to be one master-processor to ensure synchronization during integration.

It is possible to have as many processors as the number of bodies in the system, and the number of communication steps between the processors is equal to the number of neighbors of the body. If the number of neighbors is finite and does not grow according to the number of bodies in the dynamical system, then the solution time is constant. However, the architecture of the parallel processor system of a reconfigurable mesh with buses must be supposed.

The resulting system achieves oscillation with high frequencies around the solution of the original system (Fig. 3). Heterogeneous Multiscale methods [10, 11] are suitable for solving problems with high frequencies that become stiff. The solution of stiff EOM can be split into two parts, a slow part in which the solution can be depicted as a smooth curve, and a fast part in which the rapid oscillatory behavior dictates the solution. In order to re-create the original solution, therefore, the two parts must be superimposed. The working principle of heterogeneous multiscale methods consists in performing two-level numerical integrations, as shown in Fig. 4. Microintegration is carried out in the first level and consequently macrointegration is performed in the second level. The numerical integration of oscillatory EOM is realized only over a short time interval in the microintegration phase, where the small time-step is used in order to catch several periods of the oscillatory solution. The high
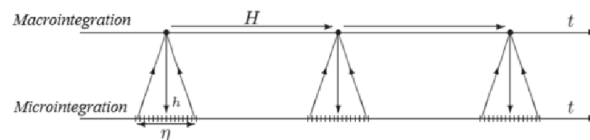

Figure 4: Scheme of multiscale integration (source: [4]).

frequencies are filtered and we obtain the smooth averaged solution of EOM, which can then be integrated with a larger time-step over the whole time interval of interest. This process is known as macrointegration.

However, the solution requires three main steps [8]. First, it is necessary to distinguish between the original system and the system with the constraints replaced by springs and dampers. Two multibody system models have been introduced in order to improve the sizes of the eigenvalues, one for microlevel and the other for macrolevel.

Second, the reaction forces have to be estimated with high accuracy in the microintegration phase. The process of double microintegration has been developed to address this issue.

Third, it is necessary to use an optimal time length of the interval of microintegration. If the interval is too short, not even one period of the oscillatory solution is captured in the microintegration. If the interval is too long, the slow averaged solution changes significantly. However, its value should be approximately constant during microintegration.

The choice of an appropriate length of η has until now been made by adaptation, but one of the next steps in this research will be to implement an automatic procedure for appropriate choice of the length of the microintegration interval.

4 LAE solution by artificial dynamics

The multibody dynamics can be solved in a different way, preserving the massive parallel structure of the computation. The EOM are assembled into the step to solve a system of LAE. Then this system of LAE is solved by additional artificial dynamics [9].

The multibody system can be uniquely described by natural coordinates, leading to the constant mass matrix. However in the case of the classical description of a body by two points B_1, B_2 on the kinematical joints and by two unit vectors v_1, v_2 (Fig. 5. A), the mass matrix has nonzero elements on its diagonal and on the second off-diagonal line over the main diagonal and under it (the matrix is symmetric). This leads to the undesired full inverse of the mass matrix. However, the mass matrix can be obtained in a pure diagonal form if an efficient set of natural coordinates is chosen, see below. Consequently the inverse of
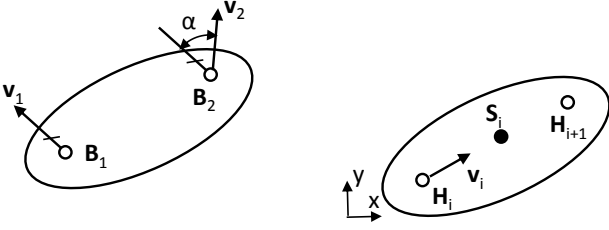
Figure 5: A body described by natural coordinates.

the mass matrix is a diagonal matrix, which is an important feature for parallel solution of multibody dynamics.

The modified state space method [7] is used in order to get the equations of motion of multibody systems. Its scheme is as follows:

The multibody system is described by a set of redundant coordinates s, which are constrained by constraint equations:

$$f(s) = 0. \tag{9}$$

The time differentiation of the constraints gives

$$J(s)\dot{s} = 0, \tag{10}$$

where matrix J is the Jacobian matrix.

Modified momentum is introduced in order to avoid the problem with constraint violation:

$$p^* = p + J^\mathsf{T}\mu = M\dot{s} + J^\mathsf{T}\mu. \tag{11}$$

Symbol p denotes the classical momentum, and μ is the vector of the modified Lagrange multipliers. If natural coordinates are used in connection with the Lagrange equations of mixed type, the following equation is obtained for the time derivatives of the modified momentum

$$\dot{p}^* = F(s, \dot{s}) + \dot{J}^\mathsf{T}\mu, \tag{12}$$

where F is the vector of the applied forces. Equation (10) is stabilized using Baumgarte stabilization and it is solved together with equation (11):

$$\begin{bmatrix} M & J^\mathsf{T} \\ J & 0 \end{bmatrix} \begin{bmatrix} \dot{s} \\ \mu \end{bmatrix} = \begin{bmatrix} p^* \\ -\alpha f \end{bmatrix}, \tag{13}$$

where α is a positive stabilization parameter. The vector of velocities $\dot{s}$ and the vector of the modified Lagrange multipliers are obtained from this equation, and consequently equation (12) is solved. Thus the time derivatives of the modified momentum $\dot{p}^*$ are obtained. The time derivatives $\dot{s}$, $\dot{p}^*$ are then integrated in order to compute the following state of the system, and the procedure is repeated.

As mentioned above, the mass matrix can be obtained in a diagonal form. It is necessary only to select an appropriate set of coordinates. A body in

a plane can be uniquely described by the Cartesian coordinates of its centre of mass S_i and by a unit vector v_i determining its orientation (Fig. 5. B). If the body is connected to other bodies with revolute joints, the mass matrix is a constant diagonal matrix, which is efficient for parallel computing.

The solution of equation (13) is the most important step from the parallelisation point of view. The Schur complement method is used to compute unknowns in a massive parallel way. It holds from (13):

$$M\dot{s} + J^\mathsf{T}\mu = p^*, \tag{14}$$

$$J\dot{s} = -\alpha f. \tag{15}$$

From (14) the vector of velocities can be expressed:

$$\dot{s} = M^{-1}(p^* - J^\mathsf{T}\mu) \tag{16}$$

and substituted in (15):

$$JM^{-1}J^\mathsf{T}\mu = \alpha f + JM^{-1}p^*. \tag{17}$$

Denoting

$$A = JM^{-1}J^\mathsf{T}, \quad b = \alpha f + JM^{-1}p^*,$$

the final equation for unknown modified Lagrange multipliers is obtained:

$$A\mu = b. \tag{18}$$

Matrix A is the Schur complement, and it is the band matrix which is symmetric positive definite for the case of natural coordinates with a diagonal mass matrix. After (18) has been solved, equation (16) is solved for known μ and subsequently the time derivatives of the modified momenta are computed (12) in order to carry out numerical time integration.

The key equation is (18), since it is necessary to solve it as fast as possible in each step of time integration. If the solution is to beindependent of the system size, it must be solved using iterative procedures. This paper describes an approach based on the idea of solving a system of linear equations as a stabilization oftostablilize the additional dynamics using numerical integration. Equation (18) is modified into the following form:

$$D\dot{\mu} + A\mu = b. \tag{19}$$

The damping matrix $D = \mathrm{diag}(D_1, D_2, \ldots, D_n)$ is diagonal with the same positive entries D_i. These values are chosen according to the diagonal values of matrix A in order to achieve suitable eigenvalues. Since matrix A is symmetric positive definite, the solution of system (19) converges to a steady state which solves the system of equations (18). System (19) is thus solved using numerical integration:

$$\mu_\eta = \mu_0 + \int_0^\eta D^{-1}\big(b - A\mu(\tau)\big)\,d\tau. \tag{20}$$

It holds for sufficient choice of η that solution $\boldsymbol{\mu_\eta}$ solves equation (18). This way of solving a system of linear equations is efficient, since it enables parallelization because matrix $\boldsymbol{D}$ is diagonal and matrix $\boldsymbol{A}$ has a band structure. In addition, the modified Lagrange multipliers change slowly during the integration, and the initial condition $\boldsymbol{\mu_0}$ is near the steady state, if it is chosen as $\boldsymbol{\mu_\eta}$ from the previous time step. Therefore it is possible to carry out the integration in (20) on many processors. Each processor can be dedicated to one body and has to communicate only with its neighbours because of band matrix $\boldsymbol{A}$. This is just the basic idea; there are other ways to allocate the processors, since they can be assigned to a group of bodies, etc.

Using these approaches, we achieved a CPU time of 19.8 s on a single processor for solving 5 pendulums, compared to 7.7 s for the traditional approach. For solving 10 pendulums, we achieved 58.0 s with these approaches, compared to 21.0 s for the traditional approach. The solution time for the method described here remains constant, while the traditional solution time grows according the computational complexity of the particular multibody formalism, i.e., at least linearly. The ratio of 2–3 between the new approach and the traditional approach means that the computational complexity will be crossed very soon (just 2–3 times more bodies), thus reaching the point where massive parallelization is an advantage.

5 Conclusion

Our paper has demonstrated that the main problem preventing efficient usage of multiple processors is the formulation of the multibody formalism, which includes solving a system of linear algebraic equations. This means that the excitation is immediately spread into all components of the multibody system. This can be avoided by introducing additional dynamics, which leads to the possibility of massive parallelization. This has been demonstrated on the basis of two approaches.

Acknowledgements

References

[1] Bhalerao, K., Anderson, K.: A recursive approach to modeling contact in flexible multibody systems. In: Arczewski, K. et al. (eds.): Proc. of Multibody Dynamics 2009, ECCOMAS Thematic Conference, Warsaw, (2009) [CDROM] pp. 1–12

[2] Binani, A. et al.: A comparison of three different linear order multibody dynamics algorithms in a limited parallel computing environment. In: Bottasso, C.L. et al. (eds.): Proc. Of Multibody Dynamics 2007, ECCOMAS Thematic Conference, Milano, (2007) [CDROM] pp. 1–21

[3] Featherstone, R.: A divide-and-conquer articulated body algorithm for parallel O(log(n)) calculation of rigid body dynamics. Part 1: basic algorithm. Int. J. Robot. Res. 18(9), (1999), pp. 867–875

[4] Featherstone, R.: A divide-and-conquer articulated body algorithm for parallel O(log(n)) calculation of rigid body dynamics. Part 2: trees, loops, and accuracy. Int. J. Robot. Res. 18(9), (1999) pp. 876–892

[5] Mukherjee, R.M., Anderson, K.S.: A robust framework for the efficient modeling and simulation of biopolymers. In: Proc. of Multibody Dynamics 2009, ECCOMAS, Warsaw, Poland (2009)

[6] Poursina, M. et al.: Energy concern in biomolecular simulations with discontinuous changes in system definition. In: Proc. of Multibody Dynamics 2009, ECCOMAS, Warsaw, Poland (2009)

[7] Sika, Z., Valasek, M.: Parallelization of multibody formalism for rigid bodies using natural coordinates and modified state space. Eur. J. Mech. A/Solids (1997), pp. 325–339

[8] Mraz, L., Valasek, M.: Solution of three key problems for massive parallelization of multibody dynamics, Multibody System Dynamics, 2012, (in print).

[9] Mraz, L., Valasek, M.: Effectivesolutionofmultibodydynamicsusingmodifiedstatespace, In: Proc. ofAppliedMechanics 2012, WBU, Pilsen 2012, CDROM pp. 1-4

[10] Calvo M. P., Sanz-Serna J. M.: Heterogenous-Multiscale methods for mechanical systems with vibrations, SIAM Journal on Scientific Computing, 2010 (vol.32) pp. 2029-2046

[11] Sanz-Serna J. M. et al.: Multiscale methods for Stiff and Constrained Mechanical Systems, Departmento de MatemáticaAplicada, Universidad de Valladolid, Spain, 2009