Current Status of Subcycling and Multiscale Simulations in LS-DYNA®

Thomas Borrvall
DYNAmore Nordic AB

Dilip Bhalsod, John O. Hallquist, Brian Wainscott
Livermore Software Technology Corporation (LSTC)

Abstract

Subcycling in explicit finite element simulations refers to the technique where a model is partitioned in levels of the characteristic time step of its constituting finite elements. Each sub model is then integrated independently of the others using a time step that pertains to that specific sub model, with the exception of special treatment at the interface between sub models. With the subcycling option in LS-DYNA, up to seven sub models are automatically generated, each integrated in steps of 1, 2, 4, 8, 16, 32 and 64 times the smallest characteristic time step of the entire model. To allow more control of the partition, the user may manually designate parts to be integrated at specific time steps. This is sometimes referred to as multiscale since it is mainly intended for detailed modeling of critical components in a large simulation model, i.e., different time scales are used in order to save CPU time. This paper presents the current status of this feature in LS-DYNA, including a detailed description of the involved algorithms and presentation of small to large scale numerical examples.

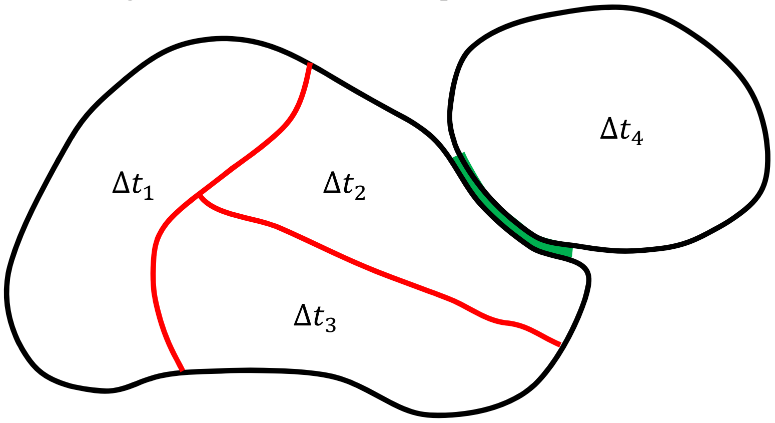


Figure 1 Using different time steps in different subdomains, interface between subdomains by constraints indicated by red and due to contact by green.

Introduction

In transient structural analysis, the stable time step for explicit time integration is bounded by the maximum natural frequency of the model. In practice the stable time step is estimated as the reciprocal of the maximum natural frequency of an arbitrary finite element, suggesting that the natural frequencies affecting stability are spatially local in their nature. To this end, it may sometimes be convenient to use multiple time steps since a local region of a structural model may require a much smaller time step than the rest, thereby decreasing the computation needed to solve the problem significantly. This method is called multi time stepping, or *subcycling*, and has been discussed extensively in the literature [2-4]. The main idea is to partition the model into sub models where each sub model is integrated with a time step that is stable for that particular region of elements, see Figure 1. Subcycling would be trivial if it wasn't for the interaction between domains that use different time steps, where the interface can be either in the form of element connections (constraints) or contacts (penalty), see Figure 1. In research a lively debate has concerned the stability and accuracy of various approaches to treat constrained interfaces, while the authors don't know of a treatment of penalty contacts in this context. The raison d'être of subcycling is the locality assumption of the natural frequency modes, but there will inevitably propagate spurious waves across interfaces between sub domains and the main question is how this will affect stability of a given method.

The present work has been undertaken under the assumption that subcycling is *in practice* stable, or at least can be made stable by combining research results and common sense. Even though LS-DYNA [1] has had a subcycling algorithm for many years this has not been used extensively for reasons pertaining to its robustness but also due to lack of its practical needs and potential gains. With the large scale state-of-the-art models in the automotive industry today, and the potential need to use ridiculously fine mesh in selected regions to accurately predict failure, the interest for the method has been revived. With this motivation, the subcycling algorithms have been revised and extended to include both *automated* and *manual* partition of the model with respect to time steps. The terminology used for distinguishing between these two ways of applying the multi time step algorithms is *subcycling* and *multiscale*, respectively. In the wake of the success of selective mass scaling, the hope is that this will provide a competitive alternative for users in need of decreasing their simulation times. The implemented algorithm is presented next and is followed by a summary of the LS-DYNA keywords. The paper ends with numerical examples and a brief summary.

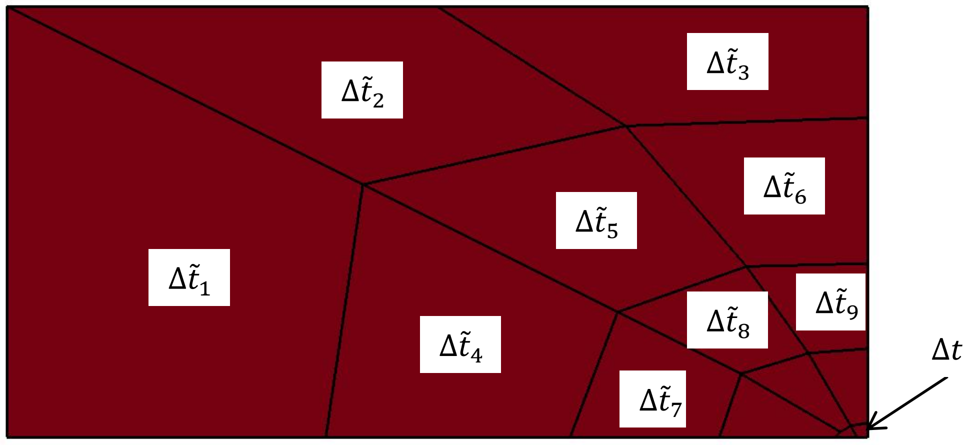


Figure 2 Example mesh for illustrating subcycling, $m = 14, n = 22$.

Subcycling algorithm

A detailed explanation of the subcycling algorithm in LS-DYNA follows, and the reader is referred to the figures for illustration of the involved steps. To simplify the exposition, we don't take into account TSSFAC on *CONTROL_TIMESTEP. Keywords that are introduced will be summarized in the next Section.

Time step calculations

A finite element model consists of m elements and n nodes, let I be the index set of all elements and J the index set of all nodes. Furthermore, let J_i be the index set of nodes for the connectivity of element i and similarly I^j the index set of elements for which node j is part of its connectivity. Each element has its *characteristic* time step $\Delta\tilde{t}_i$ calculated according to

$$\Delta\tilde{t}_i = l_i\sqrt{\rho_i/E_i} \tag{1}$$

where

l_i = Chacteristic element size (length)
E_i = Element stiffness modulus (force/area)
ρ_i = Mass density of element (mass/volume),

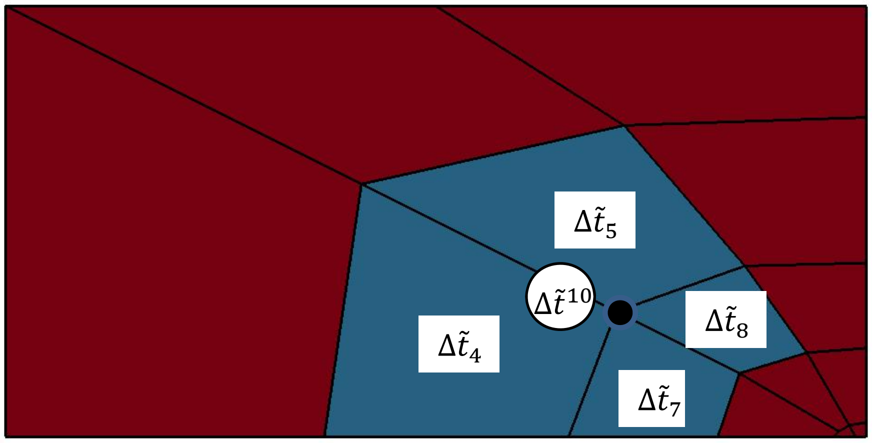


Figure 3 Characteristic nodal time step, $\Delta\tilde{t}^{10} = \min(\Delta\tilde{t}_4, \Delta\tilde{t}_5, \Delta\tilde{t}_7, \Delta\tilde{t}_8)$.

see Figure 2 for an example mesh with time steps.
The *characteristic* time step $\Delta\tilde{t}^j$ of a node is defined as

$$\Delta\tilde{t}^j = \min_{i\in I^j} \Delta\tilde{t}_i,$$

i.e., the minimum of the time steps for the elements the node is part of, see Figure 3.
The *global* simulation time step Δt is taken as the minimum of all characteristic element time steps $\Delta\tilde{t}_i$ (or equivalently nodal time steps $\Delta\tilde{t}^j$) and bounded below by a user specified time step $\Delta\bar{t}$ for mass scaling,

$$\Delta t = \max\left(\Delta\bar{t}, \min_{i\in I} \Delta\tilde{t}_i\right),$$

where in fact $\Delta\bar{t}$ corresponds to the (negative) DT2MS parameter on *CONTROL_TIMESTEP.
The *nodal simulation* time step for node j is a convenient multiple of the global simulation time step

$$\Delta t^j = k^j\Delta t \le \min\left(K\Delta t, \max\left(\Delta\bar{t}, \Delta\tilde{t}^j\right)\right) \tag{2}$$

where k^j is the largest integer of $1, 2, 4, 8, 16, 32$ or 64 that fulfils the inequality and K is an input parameter that bounds the maximum simulation time step. This is the time step that each node is using for the explicit time integration, and the keyword used for specifying the maximum time step is

*CONTROL_SUBCYCLE_{K}.

For future reference, let $\bar{K}$ be defined as

$$\bar{K} = \frac{\max_{j\in J}\Delta t^j}{\Delta t}, \tag{3}$$

noting that this number may be strictly smaller than the user defined value K.

The *element simulation* time step must not exceed the simulation time steps for the nodes in the connectivity, and hence

$$\Delta t_i = \min_{j \in J_i} \Delta t^j$$

is the time step that is used for computing internal forces, see Figure 4. It is clear that this time step will also be some multiple k_i of the global time step

$$\Delta t_i = k_i \Delta t,\ i \in I. \tag{4}$$

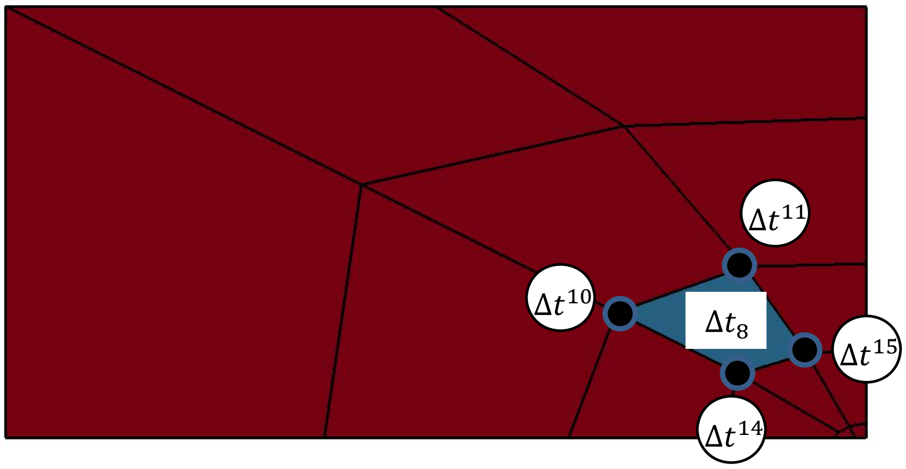


Figure 4 Nodal and element simulation time steps, $\Delta t_8 = \min(\Delta t^{10}, \Delta t^{11}, \Delta t^{14}, \Delta t^{15})$.

Forces and integration

Every $\bar{K}$:th time step is called a synchronization step, because then all nodes and elements are involved in the calculations, see (3). Assuming we are at a time t that corresponds to such a synchronization step, and for the sake of convenience that Δt is constant in time, we compute $\boldsymbol{f}_t^j$ for $j \in J$ and use this to update the coordinates at the next known point in time,

$$\boldsymbol{a}_t^j = \boldsymbol{f}_t^j / m^j$$
$$\boldsymbol{v}_{t+\Delta t^j/2}^j = \boldsymbol{v}_{t-\Delta t^j/2}^j + \Delta t^j \boldsymbol{a}_t^j$$
$$\boldsymbol{x}_{t+\Delta t^j}^j = \boldsymbol{x}_t^j + \Delta t^j \boldsymbol{v}_{t+\Delta t^j/2}^j,$$

where $m^j, \boldsymbol{a}^j, \boldsymbol{v}^j$ and $\boldsymbol{x}^j$ denote nodal mass, acceleration, velocity and coordinate, respectively. Obviously, nodal velocities and coordinates must be known at intermediate time steps and the *constant velocity* approach is used in LS-DYNA. So for node j the nodal coordinates $\boldsymbol{x}_t^j$ and $\boldsymbol{x}_{t+\Delta t^j}^j$ are known from the explicit integration scheme and intermediate coordinates are determined according to

$$\boldsymbol{x}_{t+k\Delta t}^j = \boldsymbol{x}_t^j + \frac{k\Delta t}{\Delta t^j}\left(\boldsymbol{x}_{t+\Delta t^j}^j - \boldsymbol{x}_t^j\right), k = 1, \dots, \frac{\Delta t^j}{\Delta t} - 1,$$

and the velocity is therefore constant,

$$\boldsymbol{v}_{t+k\Delta t}^j = \boldsymbol{v}_{t+\Delta t^j/2}^j, k = \frac{1}{2}, \frac{3}{2}, \dots, \frac{\Delta t^j}{\Delta t} - \frac{1}{2}.$$

The contribution to the nodal forces from elements (i.e., *internal* force contribution) is only performed when needed, from (4) it follows that an element i is processed only every k_i:th step. The *external* forces, including contacts, are processed every $\bar{L}$:th step, computed as

$$\bar{L} = \min\ (\bar{K}, L),$$

where in turn L is a user defined input parameter taken from

*CONTROL_SUBCYCLE_{K}_{L}.

If L is not specified on this keyword, $L = K$, and whence the user has the option to process contacts more or less often depending on the application. For a given node j the contribution from the external forces will be the *last computed*. This means that if $\bar{L} < k^j$ only some external nodal forces are actually used, the intermediate ones are discarded. Also, if $\bar{L} > k^j$ external nodal forces are stored and reused until new ones become available.

Multiscale

In this context, *multiscaling* is distinguished from *subcycling* in that the time steps are explicitly determined by the user. This is done by setting DT2MS on *CONTROL_TIMESTEP to a negative number and using *CONTROL_SUBCYCLE_MASS_SCALED_PART_{SET} to specify mass scaling time steps TS individually for selected parts. The outcome of this is that each element will be associated with a mass scaling time step, either one of the TS values if the element belongs to one of these parts, otherwise the (negative) DT2MS value. With reference to the previous section and abuse of notation, this replaces the *characteristic* time steps calculated in (1) by user defined *multiscale* time steps $\Delta\tilde{t}_i, i \in I$. The rest of the algorithm applies except for that $\Delta\bar{t} = \min_{i\in I} \Delta\tilde{t}_i$ and $K = 64$. An automotive application of this feature would be to integrate small but highly refined parts, like solid element spotwelds, with a small time step while the rest of the vehicle is integrated with a large time step.

Mass scaling

Regardless if subcycling or multiscaling is used, each element simulation time step will be a multiple of the global simulation time step, and as indicated in (4) this multiple integer is unique for each element. *But it must not change with time*. The reason pertains to the way LS-DYNA stores and processes elements internally and is a necessity in order to maintain computational efficiency of the algorithm. However, the characteristic time step $\Delta\tilde{t}_i$ of the elements are changing with time since the elements deform and whenever we have

$$\Delta\tilde{t}_i < \Delta t_i$$

mass scaling is used to increase the stable time step to that of the simulation time step. More specifically, the inertial mass of an element will be scaled by the factor

$$\alpha_i = \left(\frac{\Delta t_i}{\Delta\tilde{t}_i}\right)^2.$$

So, even if no mass scaling has been requested by the user, there may be mass scaling going on to render stability and efficiency to the subcycling algorithm. At this point we should also mention that the mass is rescaled only at the synchronization step, i.e., every $\bar{K}$:th step.

Contacts

Contact treatment demands a section on its own, because it is probably the most critical aspect in the subcycling framework. To get an idea of the problem, it is important to understand how penalty contacts affect stability in explicit simulations, a crude mathematical analysis follows. In principle, whenever a slave node j penetrates a master segment a *contact element* is created "on the fly", to which some contact stiffness κ_c^j is associated. In contrast to how other finite elements are treated, there is no corresponding mass added to the system and the time step is not adjusted accordingly, which means that the contact stiffness must be bounded above. If the node is associated with stiffness κ_0^j from structural elements, the accumulated stiffness in contact will be

$$\kappa^j = \kappa_0^j + \kappa_c^j, \tag{5}$$

and if the mass of the node is m^j the stable time step will decrease to that of

$$\Delta\tilde{t}^j = \sqrt{\frac{m^j}{\kappa^j}} \geq \Delta t^j. \qquad (6)$$

This should not fall below the simulation time step Δt^j to maintain stability for the contacts, and there must therefore be a margin for adding contact stiffness to the nodes. Now, since the time steps in the subcycling algorithm are determined from structural stiffness and mass, we can assume that the simulation time step Δt^j is roughly given by

$$\Delta t^j = \beta\sqrt{\frac{m^j}{\kappa_0^j}}$$

where $\beta < 1$ is a safety factor. Inserting this in the inequality (6) and using (5) in combination with (2) we get an expression for the maximum contact stiffness

$$\kappa_c^j \leq \left(\frac{1}{k^j}\right)^2 \frac{(1-\beta^2)\beta^2 m^j}{\Delta t^2} = \left(\frac{1}{k^j}\right)^2 \bar{\kappa}_c^j \qquad (7)$$

where we have used

$$\bar{\kappa}_c^j = \frac{(1-\beta^2)\beta^2 m^j}{\Delta t^2}$$

to represent a critical contact stiffness for conventional explicit simulations. From (7) we see that the contact stiffness should ideally be scaled with the square of the time step scale factor, i.e., the contact must be softer for nodes with large time steps. The challenge in this respect is how to treat interaction between nodes of different time steps; although the theoretical answer would be to use the smallest stiffness of the involved nodes this might in practice lead to a very soft contact. At the moment of writing, this issue has only been partly resolved and work is being undertaken to get a better understanding to develop a more robust contact treatment.

Keywords

The first keyword introduced in the previous Section relates to subcycling based on characteristic time steps,

*CONTROL_SUBCYCLE_{K}_{L}

and activates subcycling with a maximum ratio of K between the time step of an arbitrary node and smallest time step overall for the explicit integration scheme. K can take values 1, 2, 4, 8, 16, 32 or 64. The reason for having this parameter K as an option is mainly due to the lack of theoretical results for stability in subcycling, whence the user may determine the level at which subcycling should be made.

The parameter L should be less than or equal to K, and may also take values 1, 2, 4, 8, 16, 32 or 64. This parameter gives the maximum number of time steps between external force calculations, including (penalty) contacts. Since contacts contribute significantly to the computations it may be of benefit to use L larger than 1, but since on the other hand they also affect stability this option should be used with care. The default values of K and L are K=16 and L=1.

The second keyword relates to subcycling based on user defined time steps (multiscale),

*CONTROL_SUBCYCLE_MASS_SCALED_PART_{SET}

and when used K *is ignored* whether or not the first keyword is part of the input and regardless what the user sets K to. If the user wants to input a non-default value for L, this may be done by including the first keyword in the input. With this keyword, parts or part sets P(S)ID are associated with a given mass scaling time steps TS that indicate how to partition the model with respect to different time scales. To this end, parts that are not explicitly specified using this keyword are given the (negative) DT2MS value from *CONTROL_TIMESTEP, and whence the user is in complete control on how to subcycle the model of interest.

Examples

S-rail

An S-rail problem is solved in order to examplify the usage of the subcycling and multiscale algorithms in the context of refined spotweld modeling, see Figure 5. A spotwelded beam with added mass at the rear is given an initial velocity and impacts a fixed rigid wall, the three rail constituents consist of one fine mesh and two coarse meshes but all have the same steel material. The model consists of 24640 shell elements and 55296 solid elements, the latter ones making up the 54 spotwelds, see Figure 5 for a close-up of two spotwelds in the initial and deformed configuration. The initial smallest characteristic time step is 2.31e-5 (ms) and the problem is run on 12 processors using mass scaling corresponding to DT2MS=-2.6e-5 and a time step safety factor of TSSFAC=0.8, see *CONTROL_TIMESTEP. The simulation time of 3757 seconds for this explicit run is used as reference. By appending *CONTROL_SUBCYCLE_64_4 to the input, all solid elements are using the smallest time step while the shells are automatically partitioned into 2572/0/894/8859/10452/653/1210 elements running on 1/2/4/8/16/32/64 times the smallest time step, respectively. The simulation time for this input is 3406 seconds. As an alternative, we use the multiscale option by specifying mass scaling time steps 2.6e-5 for the solid spotwelds, 1.6e-4 for the fine shell mesh and 6e-4 for the coarse shell meshes. This leads to a partition of the model into 2572/0/20202/1866 shell elements running on 1/2/4/8 times the smallest time step, respectively. The simulation time for this input is 3454 seconds, a little higher than the subcycling case but still significantly smaller than the reference. The results are almost identical, see Figure 6, and a close examination of the timings reveal what is expected; contributions come not only from the element processing but the time spent in contact

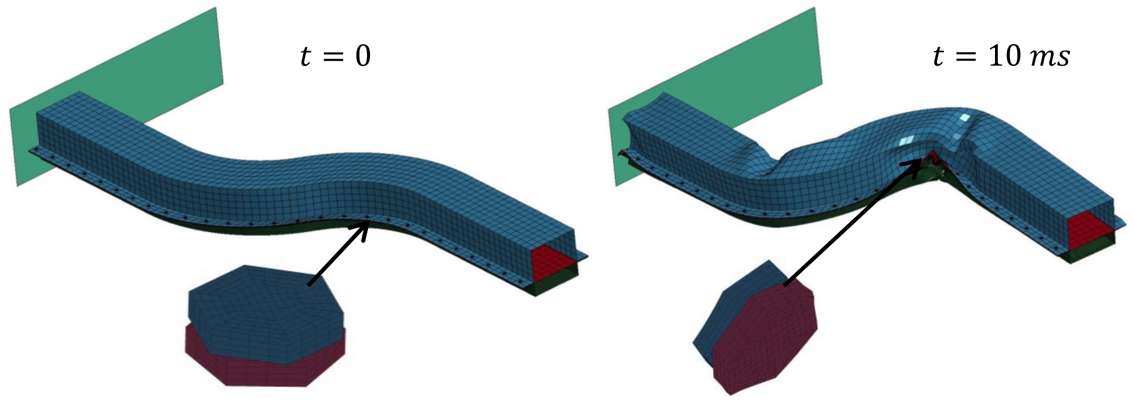


Figure 5 S-rail problem with two spotwelds highlighted.

calculations is about 4 times less for the subcycling and multiscale cases, see Table 1. This is of course due to the option L=4 for the external force calculations.

Table 1 Detailed timings for the S-rail problem.

S-rail CPU timings (s)	Subcycling, K=64, L=4	Multiscale, L=4	No subcycling
Contacts	78	79	241
Elements	1783	1790	2141

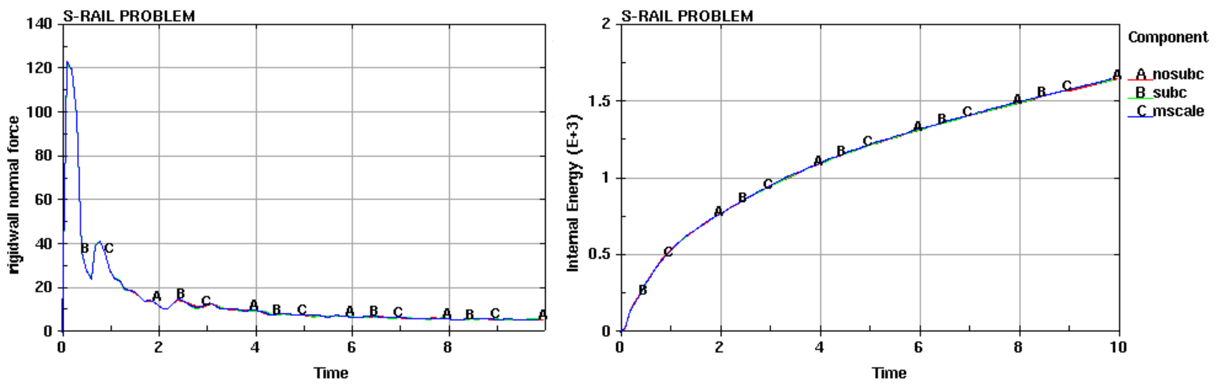


Figure 6 Rigid wall force and internal energy for the S-rail problems.

Toyota Rav4

The 1997 Toyota Rav4 model from NCAC [5] is used in a 10 ms side hit simulation, the vehicle is made up of 501587 shell elements and 21539 solid elements. A 7 kg cylinder impacts the side with a velocity of 30 mph (13.4 m/s) at the location of the B-pillar, see Figure 7 for the initial and final configuration. In the B-pillar 2 parts are refined to demonstrate the use of subcycling and multiscale, and the smallest initial time step is 1.3e-4 (ms) for one of these refined elements. On *CONTROL_TIMESTEP, DT2MS=-1.249e-4 is used for mass scaling of the model, this simulation takes 1443 seconds on 48 processors. As for the S-rail problem, *CONTROL_SUBCYCLE_64_4 is used for the subcycling run which leads to a partition of the vehicle according to Figure 8. The B-pillar, rigid materials and elements connected by spotwelds (nodal rigid bodies) use small time steps while most of the vehicle is using 8 times the smallest time step size or higher. The multiscale option is then used by using a time step of 1.249e-4 for the B-pillar parts indicated in Figure 7 and a DT2MS=-1e-3 for the rest of the vehicle, this changes the partition of the model into 93948 shells and 1838 solids using the smallest time step (the B-pillar, rigid materials and spotweld connections), while 407639 shells and 19701 solids use 8 times the smallest step.

The subcycling and multiscale simulations take 795 and 906 seconds, respectively, so for the Rav4 this technique is really paying off. The detailed timings in Table 2 show similar tendencies as for the S-rail problem, and results are insensitive to applying either subcycling or multiscale, see Figure 9. However, when comparing these timings to applying selective mass scaling on the B-pillar parts and using DT2MS=-1e-3 it turns out that both methods are behind, the selective mass scaling simulation completes in 304 s with very similar results. We hesitate to draw general conclusions from this since this example may not represent the best possible usage of either of the methods. What would be the outcome if for instance the model size was to be increased or if the refined parts were to be distributed all over the vehicle, as it would for deformable spotwelds? More questions like these may be asked but only speculative answers can be made, the definite ones lie in the future.

Table 2 Detailed timings for the Toyota Rav4.

Rav4 CPU timings (s)	Subcycling, K=64, L=4	Multiscale, L=4	No subcycling
Contacts	133	133	288
Elements	194	206	636

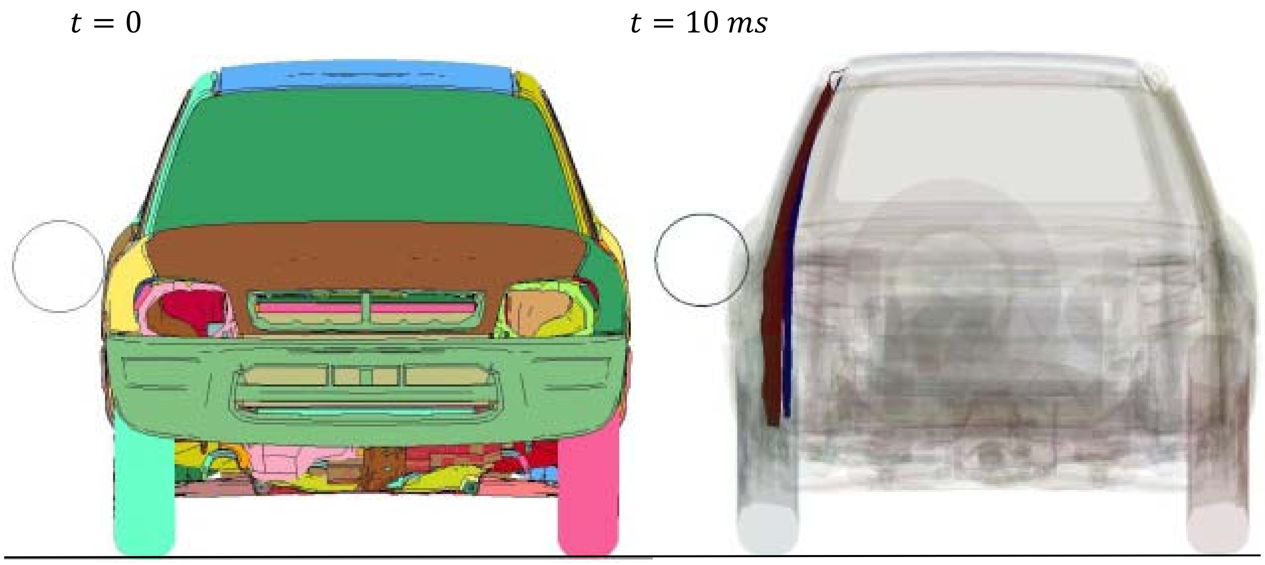
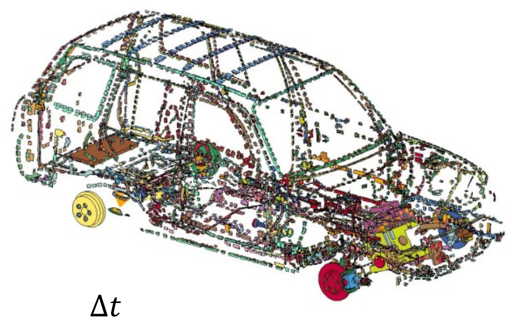


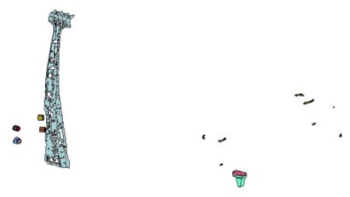
Figure 7 Cylinder hitting side of the Toyota Rav4, refined B-pillar parts.

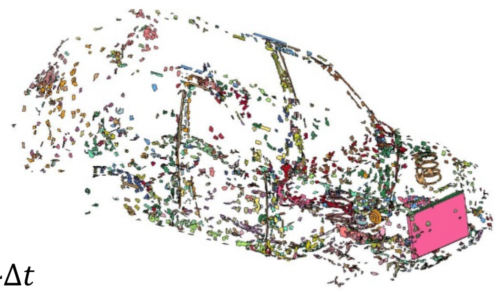
Toyota Camry

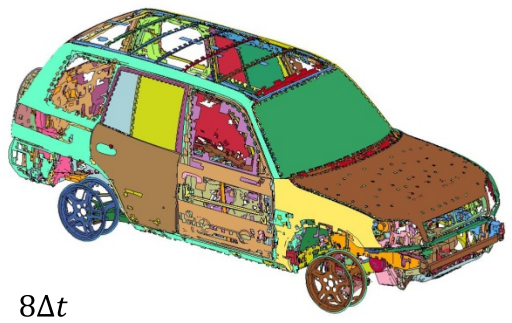
The latest NCAC model is a Toyota Camry, consisting of 1603439 shell elements, 64257 solid elements and 5102 beam elements, that impacts a stationary rigid wall at a speed of 35 mph (56 km/h), see [6] and Figure 10. A few model adjustments were made to improve robustness, and from that a standard explicit simulation is compared to that of using subcycling (no multiscale). Out of a few attempts with subcycling, some terminated with errors, some terminated successfully but with poor results and some terminated successfully with acceptable results. The experience from running this example is that contacts play an important role in getting subcycling robust and accurate, and adjustments of the contact stiffness usually has an effect of the outcome. The original model comes with a segment-to-segment based single surface contact algorithm for the vehicle (SOFT=2 on *CONTACT_AUTOMATIC), but the common node-to-segment contact (SOFT=0 on *CONTACT_AUTOMATIC) is also used in our tests. The results in this section come from using the latter contact with default contact stiffness and the default settings for subcycling, i.e., *CONTROL_SUBCYCLE_16_1. The model is run with a mass scaling time step of 1 microsecond, and as it turns out, see Figure 11, 91 % of all elements are associated with the smallest time step and unfortunately great savings in CPU timings cannot be expected. This model is included herein mostly to examine the robustness of the subcycling algorithm for a common large scale automotive application. Comparison of CPU timings between the original and subcycling runs is found in Table 3 and results in form of energy and rigid wall force is shown in Figure 12.

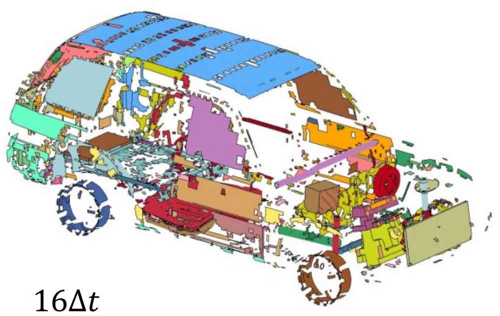

Δt

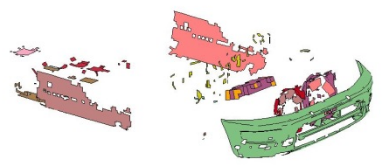
Δt	66899 shells	1838 solids
$2\Delta t$	24685 shells	295 solids
$4\Delta t$	36768 shells	9424 solids
$8\Delta t$	290160 shells	2869 solids
$16\Delta t$	60876 shells	7007 solids
$32\Delta t$	9684 shells	45 solids
$64\Delta t$	12515 shells	61 solids


$2\Delta t$


$4\Delta t$


$8\Delta t$


$16\Delta t$


$32\Delta t$

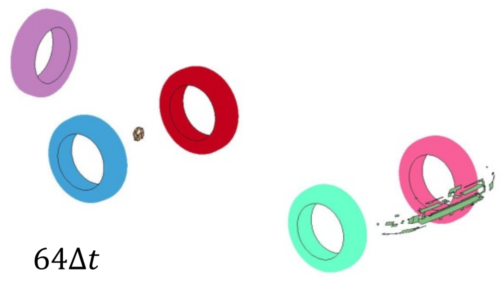

$64\Delta t$

Figure 8 Subcycling partition of the Toyota Rav4.

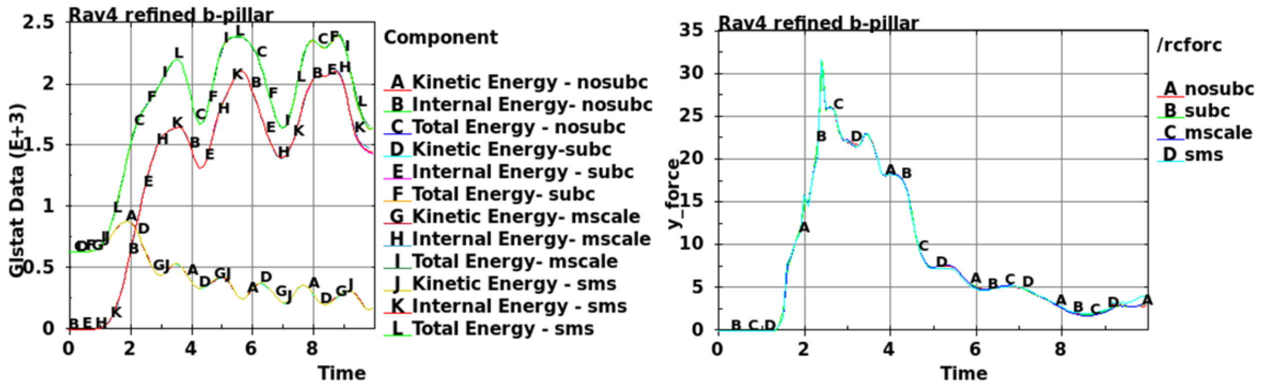


Figure 9 Energies and cylinder contact force for the Toyota Rav4 side hitting simulations.

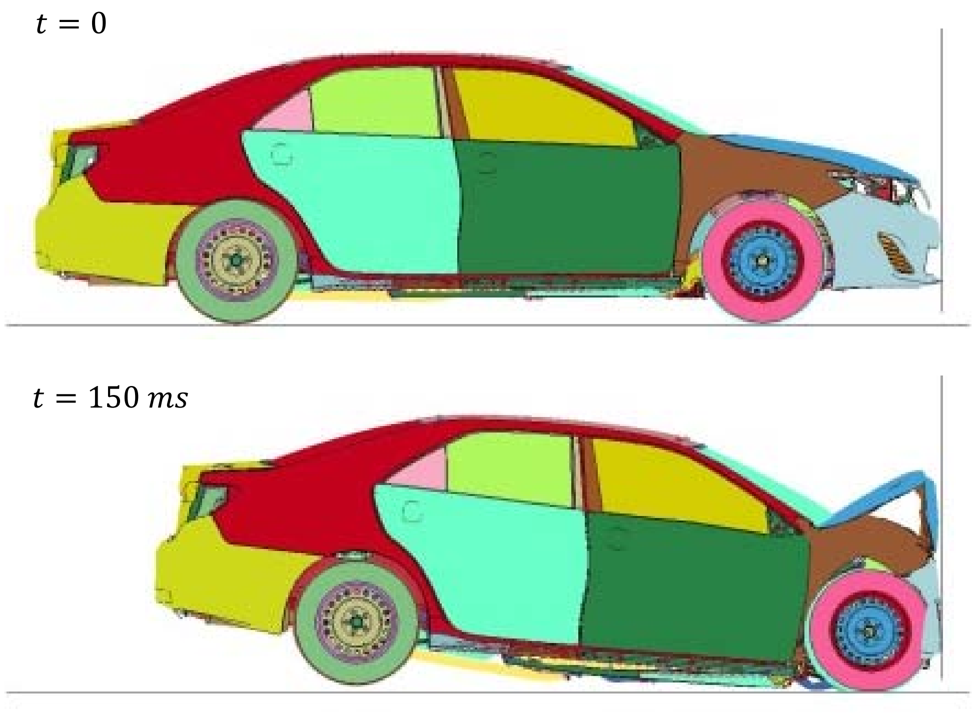
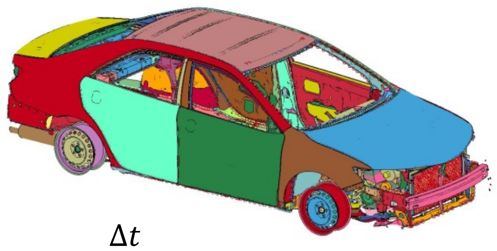
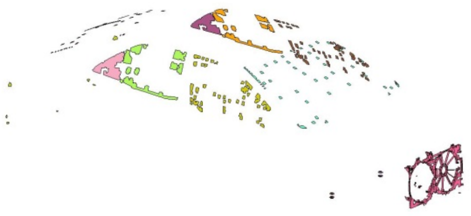


Figure 10 Frontal impact of the Toyota Camry.

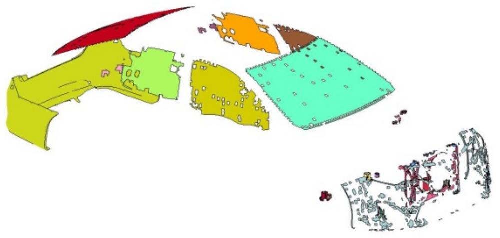


Δt

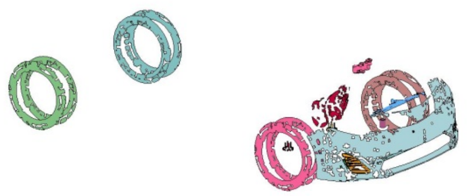
Δt	1460807 shells	61541 solids
$2\Delta t$	10352 shells	48 solids
$4\Delta t$	62167 shells	1440 solids
$8\Delta t$	38449 shells	1228 solids
$16\Delta t$	31618 shells	0 solids



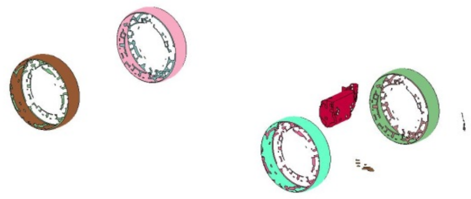
$2\Delta t$



$4\Delta t$



$8\Delta t$



$16\Delta t$

Figure 11 Subcycling partition of the Toyota Camry.

Summary

Revised subcycling and multiscale algorithms implemented in LS-DYNA have been described in detail. The results from numerical examples indicate that there is a potential of reducing simulation time without significantly affecting the results, which is a motivation for further development. The near future will be devoted to understanding and improving contacts in subcycling, as this currently seems to be the most critical factor when it comes to stability of the method. In our only comparison of subcycling to that of selective mass scaling, see the Toyota Rav4 example, the latter came out as a winner. However, we don't claim that the presented examples reflect the best use of the method but should be viewed as teasers, more investigations would be needed to find the appropriate application. Closely related to this is the question how subcycling and selective mass scaling will stand the test of time; the endless growth of finite element models and number of processors used for simulating them. The answer lies in the unforeseen future and it's too early to draw definite conclusions, but we hope that the content of this paper will inspire LS-DYNA users to investigate the potential of the method within their own specific field.

Table 3 CPU timings for the Toyota Camry (in seconds).

#CPU	No subcycling	Subcycling
12	28942	28091
24	14964	14158
48	7578	7642
96	4122	4028

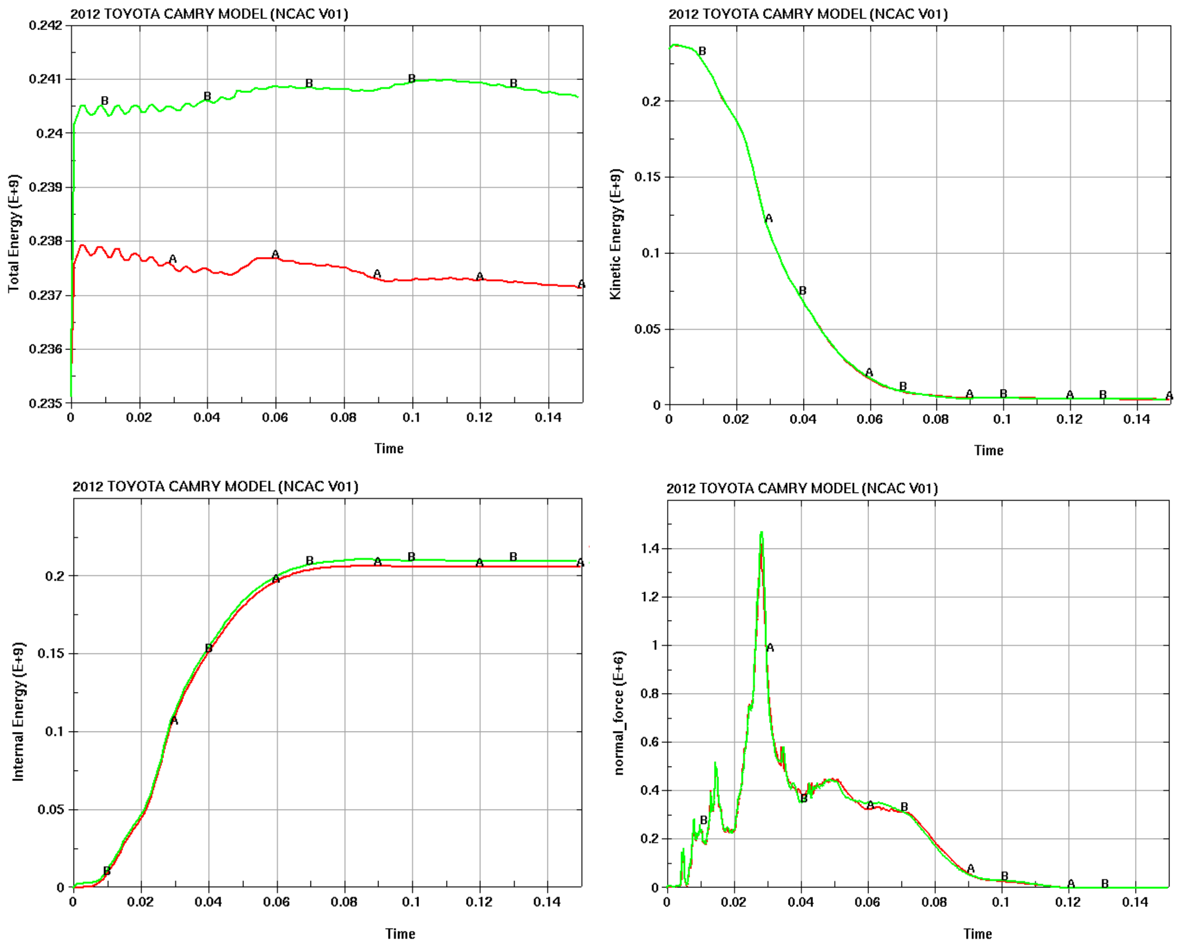


Figure 12 Total, kinetic, internal energy and rigid wall force for the Toyota Camry using no subcycling (A) and subcycling (B).

References

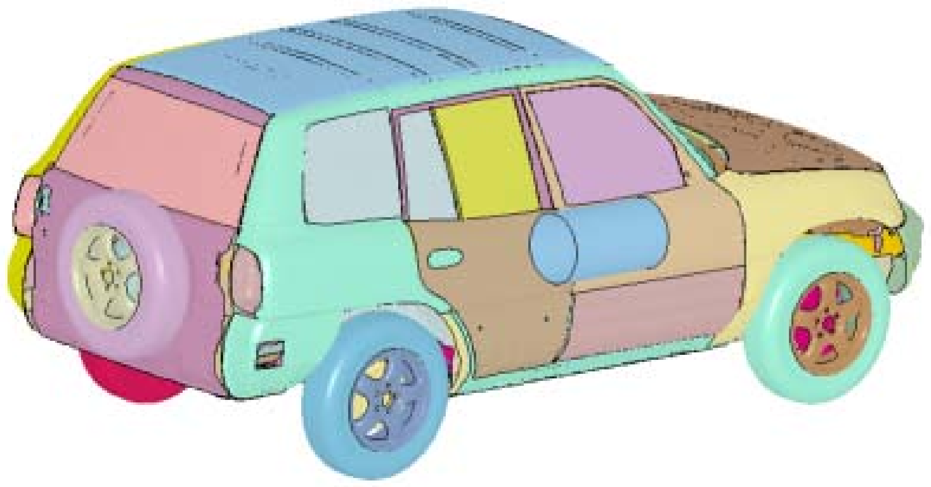
[1] LS-DYNA Keyword User's Manual, Vol I-II, Livermore Software Technology Corporation (LSTC), 2013.

[2] T. Belytschko and R. Mullen, *Explicit integration of structural problems*, in P. Bergen et.al. (eds.), Finite Elements in Nonlinear Mechanics, Vol. 2, (1977), pp. 697-720.

[3] T. Belytschko and Y.Y. Lu, *Explicit multi-time step integration for first and second order finite element semidiscretizations*, Computational Methods of Appl. Mech. and Engrg., 108 (1993), pp. 353-383.

[4] P. Smolinski, S. Sleith and T. Belytschko, *Stability of an explicit multi-time step integration algorithm for linear structural dynamics equation*, Computational Mechanics, 18 (1996), pp. 236-244.

[5] 1997 Toyota Rav4, Detailed Model (494,117 elements)
http://www.ncac.gwu.edu/vml/archive/ncac/vehicle/rav4-v1.tgz
http://www.ncac.gwu.edu/vml/archive/ncac/vehicle/rav4-v1.pdf



[6] 2012 Toyota Camry, Detailed Model (1,672,758 elements)
http://www.ncac.gwu.edu/vml/archive/ncac/vehicle/Camry-v01.tgz
http://www.ncac.gwu.edu/vml/archive/ncac/vehicle/CamryV1_Report.pdf