Increasing Processing Speed for Interactive Real-Time Simulation Environments

Ian Flood
University of Florida
Gainesville, FL 32611-5703
flood@ufl.edu

Raja R.A. Issa
University of Florida
Gainesville, FL 32611-5703
raymond-issa@ufl.edu

Caesar Abi Shedid
Florida International University
Miami, FL 33174
cabi@fiu.edu

Abstract

The paper is concerned with simulating physical processes in buildings (such as heat transfer, fire propagation, and dynamic stress response within a structure) at real-time and accelerated-time speeds, so that they may be used in an interactive 4 dimensional visualization environment. Initially, it is demonstrated that conventional computing techniques will not be able to achieve satisfactory processing speeds within our lifetime, since their rate of progress is overwhelmed by the size of these models. Alternative computing techniques are then explored as means of achieving the required processing speed, including the use of parallel computers and direct mapping models. It is shown that the only current technology that has any potential of resolving this problem is the coarse-grain method (CGM). CGM is briefly introduced and a summary of its performance capabilities are presented for the problem of modeling transient heat-flow in buildings. The paper concludes with an identification of where future research needs to be focused..

Keywords

Accelerated Time Simulation; Artificial Neural Networks; Coarse-grain modeling; Parallel Computing; Real-Time Simulation; Visualization

1. INTRODUCTION

Virtual reality and related environments usually require the representation of dynamic visual information in real-time or accelerated time. Today's computers can readily handle the processing involved with modeling such environments provided the image only has to adjust in response to a change of position/orientation of the viewer, or in response to similar simple shifts in the orientation of the environment or its objects. A problem arises, however, where changes in the environment must be generated in real-time or faster using numeric simulation techniques. Unfortunately, there are many situations within the A/E/C disciplines where this is the case. Examples include modeling fire and smoke propagation through buildings (NIST, 2006), modeling transient heat-flow through structures (US Department of Energy, 2006), and visualizing the distribution of stresses throughout a structure (UC Berkley, 2006). In all cases, visualizing these phenomena in a virtual environment would help determine the effectiveness of alternative designs, evaluate the impact on building performance of alternative design decisions and, in the case of fire propagation, help emergency crews determine appropriate strategies for fire fighting and evacuation purposes. Typically, a three-dimensional simulation of such problems can take several days or even weeks to process, making interactive visualization impossible. Even problems where visualization is required to occur at less than real-time speeds (such as the propagation of blast waves through a building) cannot be processed fast enough since the simulation operates orders of magnitude slower than the required viewing speed.

Numeric models are inherently expensive in terms of computer processing time (see, for example, Chen et al., 2000). They require the system under investigation to be represented by a very large number of elements, each representing the state of the system at a discrete location in the modeled space. The state of the model is advanced in small time increments by resolving a set of driving equations for each element, derived from known physical laws. The speed of execution of a simulation is thus dependent on the number of elements in the model and the size of the time steps. Generally, the accuracy of the simulation improves as the sizes of the spatial elements and the time increment are decreased. However, reducing the size of the spatial elements results in an increase in their total number, and consequently an increase in the processing time (this is a geometric rate of increase for models operating in two or three spatial dimensions). Similarly, reducing the size of the time increments increases the number that must be executed in order to advance the model over a given period of time, further increasing the amount of processing to be executed.

As a practical example, consider the problem of modeling the flow of heat through a building. Research indicates (Abi-Shdid, 2005) that this type of model can

achieve an acceptable level of accuracy using element sizes of around 50 mm (which is relatively coarse compared to other problems, such as stress analysis). Thus, a small structure 5 meters cube would require in the order of 1 million spatial elements. Doubling the structure size to 10 meters would increase the number of elements required eight-fold to around 8 million spatial elements. The research also shows that the relatively long time step of around 1 minute is sufficient to produce accurate predictions for this type of problem. This results in approximately a half million time steps for a one year simulation. For the 5 meter cube structure, this would require solving the driving equations $5x10^{11}$ times (half a trillion) in the simulation of a year of heating and cooling. This would take about 2.74 years to process on a typical desk top computer. For two-dimensional modeling the situation is not so critical, however most buildings cannot be modeled satisfactorily in two spatial dimensions.

There are several ways in which this problem might be overcome. These are:

i. Wait for the on-going advances in computer hardware performance to reach a stage whereby real-time and accelerated-time execution of such simulations becomes feasible.
ii. Use parallel computer technology to increase the speed of execution of simulations.
iii. Use empirically derived models that can map directly from a description of the problem to its state at a specified point in time, thereby making processing time largely independent of model complexity.
iv. Use empirically derived models in a coarse-grain modeling environment, substituting spatial sampling for temporal sampling, thus reducing the effective size of the model.
v. Any combination of the above.

The objective of this paper is to assess each of these approaches as a possible solution to the problem at hand, identifying their potential and their relative advantages and disadvantages.

2. ADVANCES IN CONVENTIONAL COMPUTING HARDWARE

Moore's Law (Intel, 2006) states that computing power (including processing speed) can be expected to double approximately every two years. Given such a geometric rate of increase in performance, it might be expected that very soon computers would be sufficiently powerful to make the speed of execution of a simulation a non-issue. However, counteracting this is the fact that the number of spatial modeling elements in a model is a cubic function of the size of the system under investigation. Taking these factors into account, as shown in Figure 1, it is possible to predict when computers will be sufficiently powerful to execute a given model at an acceptable rate for a virtual reality implementation.

This is illustrated in Figure 1 which shows the year in which we can expect desktop computers to be fast enough to model transient heat-flow in structures of different size, using 50 mm sized elements and a 1 minute time increment (performance is based on measurements reported in Flood et al (2004)). Each curve in the figure represents different rates of execution: (i) real-time; (ii) accelerated time whereby a year is simulated in a day; (iii) accelerated time whereby a year is simulated in an hour; and (iv) accelerated time whereby a year is simulated in a minute. A 10 cubic meter structure, for example, could not be executed at real-time until the year 2013. In an interactive visualization environment, it is more likely that we would require the simulation to run at the rate of one year of simulated time per hour of actual time, which could not be achieved until the year 2039. For larger structures, the problem is significantly worse. Clearly, such improvements in computing power are not going to be sufficient within our lifetime. Moreover, there is no guarantee that computing speed will continue to improve at this rate very far into the future.

3. PARALLEL COMPUTING

An alternative approach to increasing processing speed is the use of parallel computing hardware (Elsevier, 2006). A parallel computer is a device comprising two or more processing units that operate simultaneously. The idea is that the processing load can be divided between these units and thus completed at a much faster rate. Thus, if we were to allocate one processor to each spatial element in a model we might expect the rate of execution of a simulation to be independent of the size of the model. A problem with parallel computers, however, is the need for the processors to communicate with each other during a simulation which leads to significant overheads in processing. Consequently, a device comprising '*n*' processors will not execute a problem '*n*' times faster than a single processor. For simulations of the type considered here (modeling the behavior of physical processes in a spatial-temporal framework) this problem may not be too serious since each processor will only need to communicate with its immediate neighbors, as illustrated in Figure 2.

However, the parallel computing approach has a similar drawback to the previous solution (waiting for computers to get faster) in that while the number of processing elements available on a given integrated circuit may increase exponentially (perhaps doubling every two years), the number of processing elements required to run a simulation will increase as a cubic function of the size of the structure being modeled. Consequently, it will be many years before parallel computers with sufficient numbers of processors will be available to run all but the simplest of simulation models. Moreover, little effort is

being made currently to develop this type of technology for desktop computers, and it is likely this will not change until performance improvements in single processing devices start to reach a plateau.

4. DIRECT MAPPING

The third approach listed for achieving simulation speeds suitable for interactive real-time and accelerated-time visualization environments is that of direct mapping from one state of the model to another. This requires the development of the mapping function using some empirical modeling tool, such as regression analysis or artificial neural networks, based on observations of the behavior of the system.

The advantages of a mapping function approach are that: (i) the model can be developed to advance by very long time increments (where required), thus skipping uninteresting periods in the simulation, and (ii) that the mapping function typically involves a lot less processing than solving the driving equations in a numeric model. However, the number of inputs required to a mapping function will increase directly with the number of elements that would be required in a conventional numeric model (that is, it will increase cubically with the size of the model under investigation), which will cause a corresponding cubic increase in the amount of processing to be performed. (Note, a mapping function for a large model could be used to simulate a smaller system by setting the unused inputs to a null value).

A more critical problem is that the number of observations required to develop a mapping function increases geometrically with the number of inputs. Consequently, it is only possible to develop such functions for very small models since the number of observations required and the model development time quickly become unwieldy.

5. COARSE-GRAIN METHODS

The final approach proposed here is to use a coarse-grain method of modeling (CGM), which decomposes a structure into a small number of high-level elements (Flood et al., 2004). To ensure that the accuracy of predictions is not compromised by the coarseness of the spatial sampling in this approach, each element uses historic sampling of the recent states of its immediate environment as predictors. The functions that map from the predictor to the new state of an element are likely to be multivariate, nonlinear in form, and essentially unknown, thus they must be developed using an appropriate empirical modeling technique such as that of artificial neural networks (ANNs). In this sense, coarse-grain modeling is an extension of the direct-mapping approach discussed in the previous section, but without the problem of a geometric increase in the complexity of the model with an increase in the size of the system under investigation.

The end-user of such a CGM modeling system would be provided with a comprehensive menu of modeling element types from which each new model would be configured. Each element would include a set of attribute variables which would enable the user to define the specific characteristics of the component for the system under investigation. The use of attribute variables, as such, has the added advantage of simplifying experimentation with a CGM model since many alternative building designs can be considered by simply changing the value of a variable rather than reconfiguring the modeling elements.

A proof of concept of this approach has been considered in an earlier study for modeling transient heat-flow in buildings (Flood et al., 2004). In this application, each coarse-grain modeling element represents either a space in the building (such as a room or attic) or a solid boundary between spaces (such as a wall or floor), as illustrated in Figure 3. The low spatial resolution of this approach was found to provide a rapid execution of a simulation, several orders of magnitude faster than the conventional numeric modeling approach, with a reasonable degree of accuracy (to within 2.3 ^{o}F by the end of a simulated year).

Based on this rate of execution of a simulation, Figure 4 shows how long it will be before the coarse-grain approach can achieve: (i) accelerated time whereby a year is simulated in a day; (ii) accelerated time whereby a year is simulated in an hour; and (iii) accelerated time whereby a year is simulated in a minute. Comparing this to Figure 1, it is clear that the coarse-grain approach can achieve the simulation speeds required for real-time and accelerated-time visualizations in a much more acceptable period of time than conventional numeric simulation methods.

The disadvantage of the coarse-grain approach is that a comprehensive library of coarse-grain elements must be established, although this can be done in the laboratory and is not the concern of the end-user. In addition, this is a new approach to modeling and has so far only been tested on the problem of transient heat-flow in buildings. Its ability to model accurately other physical phenomena, such as fire propagation in buildings, is yet to be tested.

6. CONCLUSIONS

The paper has discussed the problem of achieving sufficiently fast simulation of engineering processes to permit interactive visualization. First, it was demonstrated (for heat-transfer problems) that conventional computing technology (single processor digital devices) will not be sufficiently powerful to achieve real-time processing speeds in our lifetime, since the rate of progress of this technology is overwhelmed by the size of the problem. Parallel processing was also shown not to be a feasible solution in the foreseeable future for similar reasons. The alternative approach of direct mapping is also not

viable since the number of observations required to develop a model increases geometrically with the size of the model.

The most promising method is that of coarse-grain modeling which is in essence a development of the direct mapping approach. Previous studies have demonstrated the approach and its feasibility for modeling transient heat-flow in buildings. Future work needs to focus on developing and validating this approach for other applications such as fire and smoke propagation in buildings and dynamic stress distribution in structures.

7. REFERENCES

Abi-Shdid, C., (2005), "ANN Coarse-Grain Based-Model for Simulating Two-Dimensional Heat Transfer in Buildings", PhD. Dissertation submitted in partial fulfillment for the degree of Doctor of Philosophy, University of Florida.

Chen, C J, Bernatz, R., Carlson, K.D., and Lin, W., (2000), "Finite Analytic Method in Flows and Heat Transfers", Taylor and Francis, New York.

Elsevier, (2006), Journal of Parallel Computing, http://www.elsevier.com/wps/find/journaldescription.cws_home/505617/description#description

Flood, I., Issa, RRA., and Abi-Shdid, C., (2004), "Simulating the Thermal Behavior of Buildings Using Artificial Neural Network-Based Coarse-Grain Modeling", *Journal of Computing in Civil Engineering*, ASCE, 18 (3), pp 207-214.

Intel, (2006), "Excerpts from a Conversation with Gordon Moore: Moore's Law", http://en.wikipedia.org/wiki/Moore's_law

NIST, (2006), The Fire Dynamics Simulator, http://fire.nist.gov/fds/

University of California, Berkeley, (2006), Earthquake Engineering Research Center Software, http://nisee.berkeley.edu/software/

United States Department of Energy, (2006), Building Energy Software Tools Directory, http://www.eere.energy.gov/buildings/tools_directory/

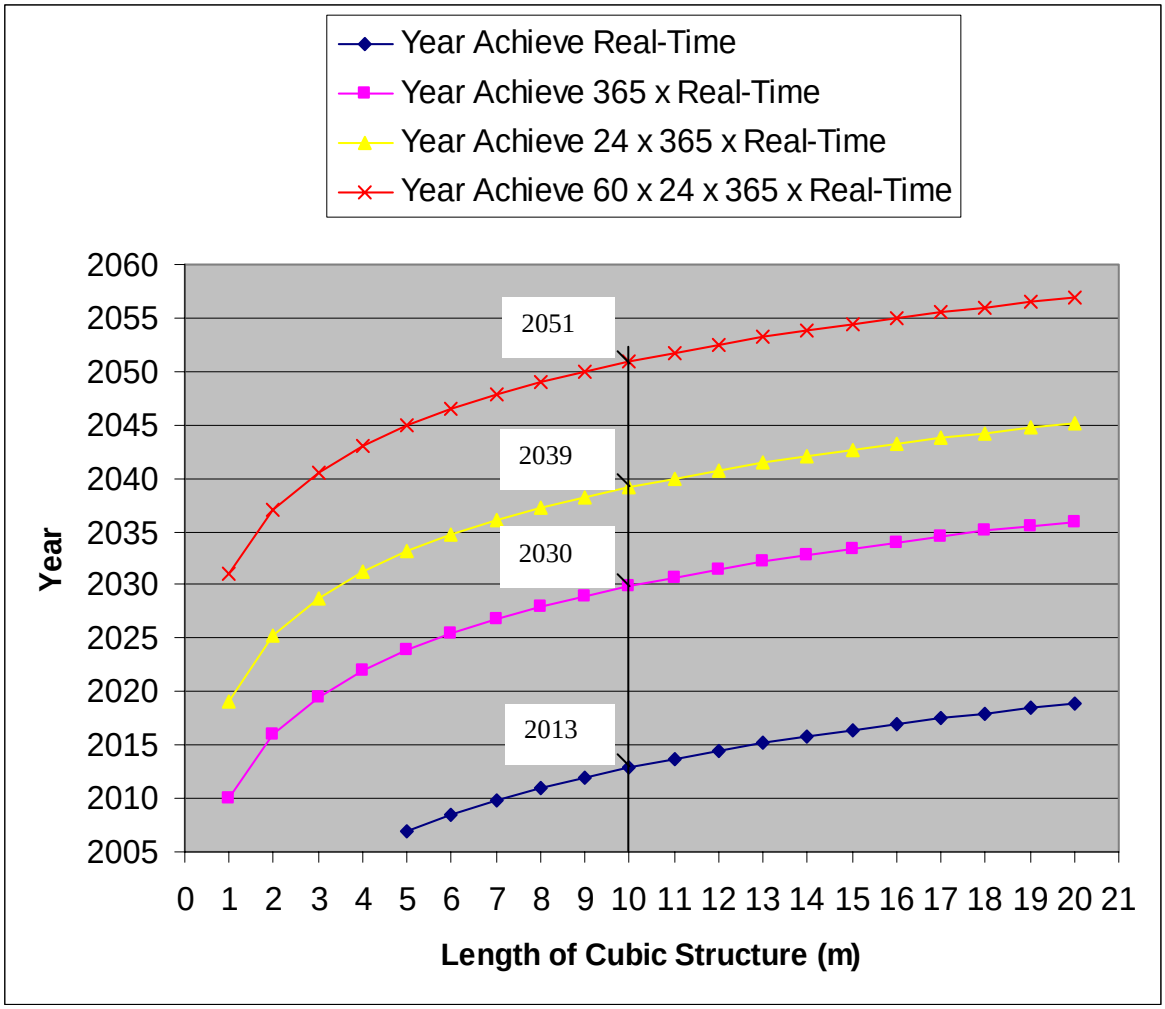
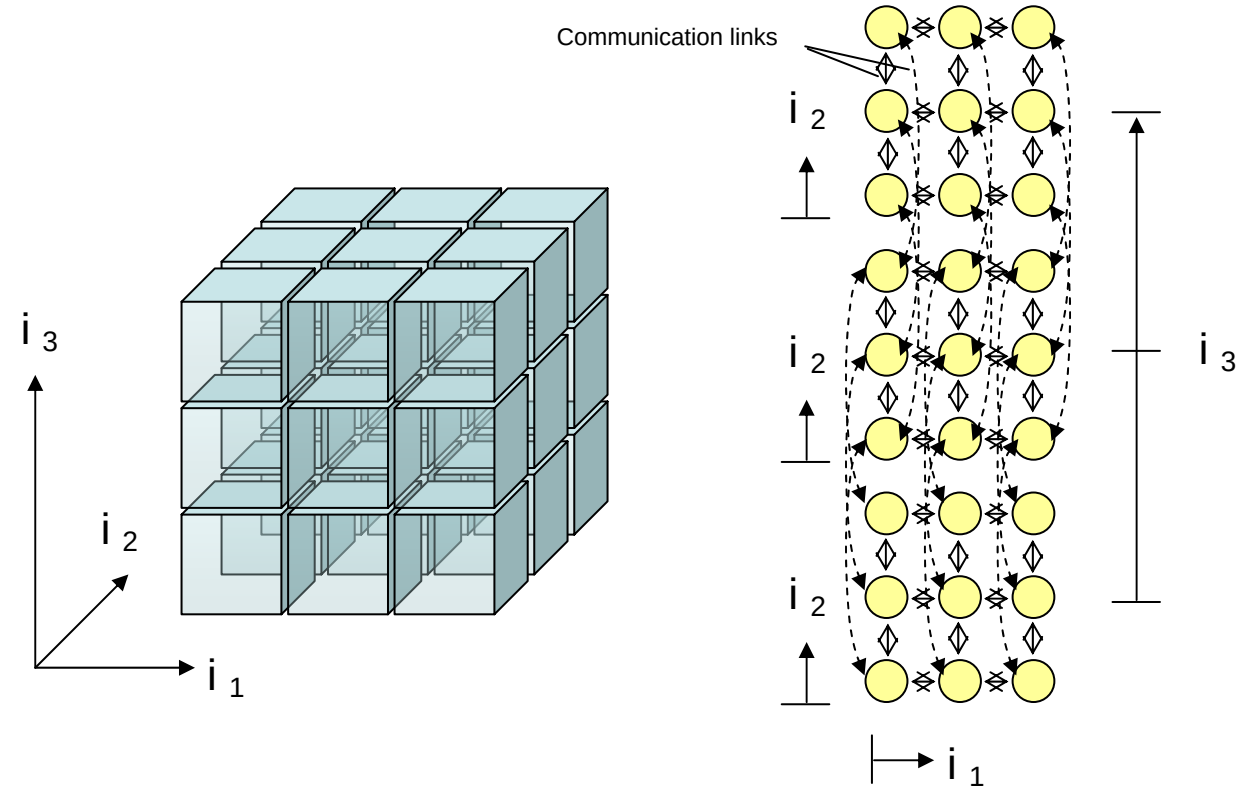


Figure 1: Year when desktop computers can be expected to have an acceptable rate of processing for heat-transfer simulation models of varying size



(a) Simple 3 x 3 x 3 Element Cubic Model (b) Parallel Processors

Figure 2: Mapping of spatially distributed modeling elements onto a parallel processor.

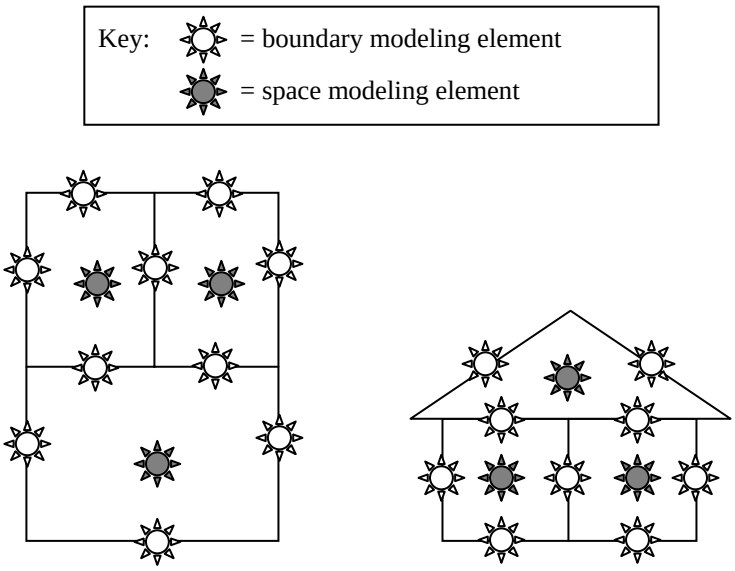


Figure 3: Assemblage of Coarse-Grain Modeling Elements for a Simple Structure

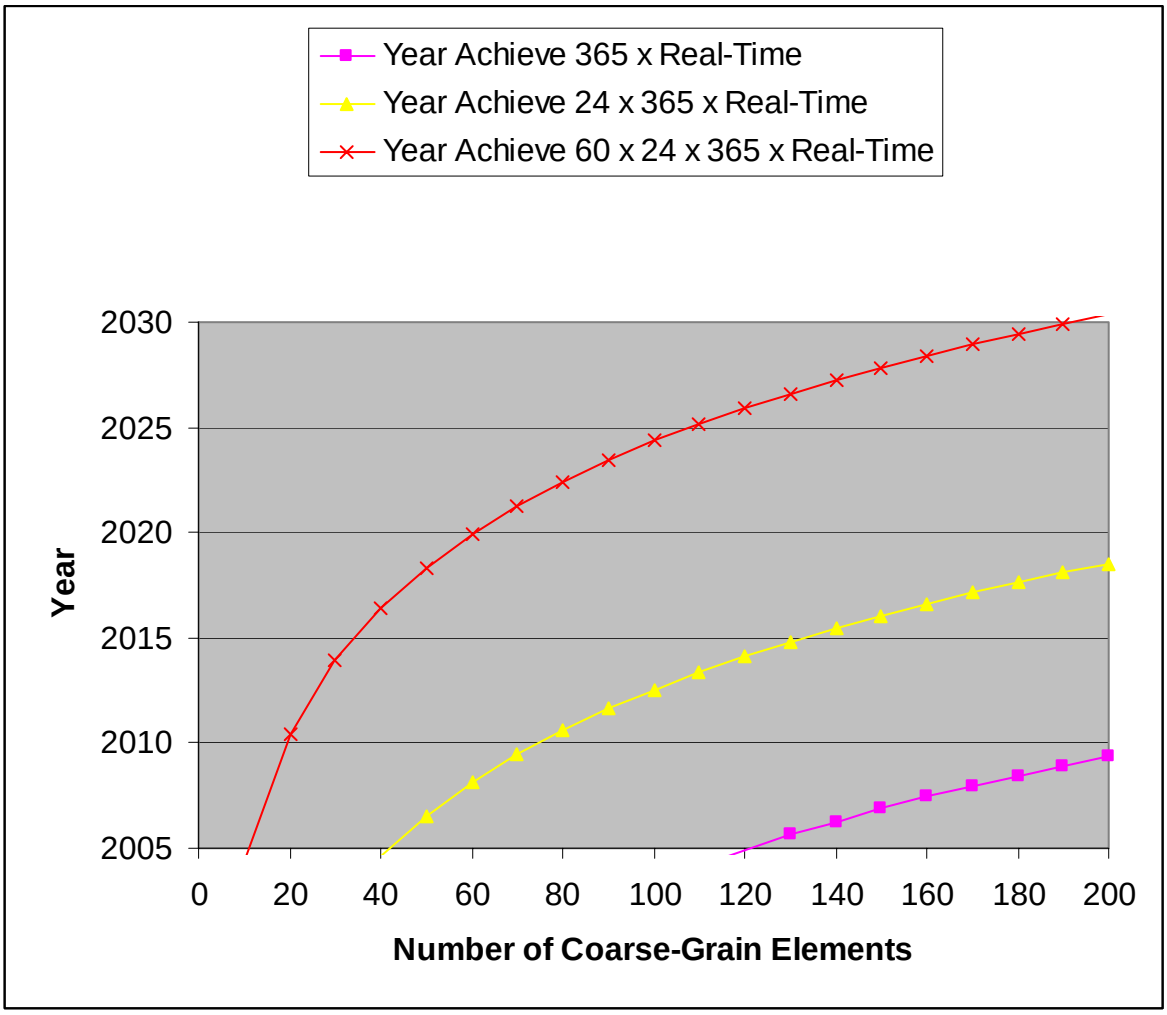


Figure 4: Year when desktop computers can be expected to have an acceptable rate of processing for heat-transfer simulation models comprising various numbers of coarse-grain elements