ICT-CELLULAR-COMBUSTION-ALGORITHM (ICCA)

APPLICATION TO THE COMBUSTION OF COMPLEX SHAPED PROPELLANTS

Sebastian Wurster, Thomas S. Fischer

Fraunhofer Institute for Chemical Technology, Joseph-von-Fraunhoferstr. 7, D-76327 Pfinztal-Berghausen, Germany

Corresponding Author: sebastian.wurster@ict.fraunhofer.de

Abstract

The ICT-Cellular-Combustion-Algorithm (ICCA), an algorithm that uses an approach based on cellular automata to calculate the geometry changes during combustion of solid propellants, is presented. The theoretical foundations of the algorithm are explained, the creation of input data is shown and the connection to the simulation of propellant combustion is outlined. The algorithm can be further used to create an animated visualization of geometry changes during propellant combustion. Finally challenges and possible solutions in the application of ICCA to the simulation of complex shaped solid propellants are discussed.

Keywords: Propellant, Combustion, Simulation, Modeling, ICT-CELLULAR-COMBUSTION-ALGORITHM, ICCA, Interior Ballistics, Gun, Cellular Automata

Introduction

The three main factors that have to be known to simulate the combustion of a solid propellant are the thermodynamic properties of the product gases that are created during combustion, the pressure dependent linear burning rate $de/dt = r(p)$ and the form function $\varphi(z)$.

The gas production rate can be calculated from the linear burning rate $de/dt = r(p)$, most often found in the form of Vielle´s law and determined in closed vessel experiments. In addition the form function φ as a function of the burned mass fraction z has to be known. The form function, defined as the burning surface area over the initial surface area $\varphi(z) = S/S_0$ of

the solid propellant, encodes the geometry changes during combustion and has to be known as an analytic function in most interior ballistic codes, e.g. IBHVG-Code [1]. The linear burning rate de/dt, the form function φ(z) and the gas production rate dz/dt are linked as stated in formula (1).

$$\frac{dz}{dt} = \left(\frac{S_0}{V_0}\right) \cdot \left(\frac{S}{S_0}\right) \cdot \frac{de}{dt} = \left(\frac{S_0}{V_0}\right) \cdot \varphi(z) \cdot r(p) \tag{1}$$

The thermodynamic properties of the reaction products can be either measured experimentally using closed vessel data or calculated using a thermodynamic code, e.g. the ICT-Thermodynamic-Code [2]. Using an equation of state (EOS), e.g. the Virial-EOS, the pressure p can be linked to the burned mass fraction z as a function of time.

$$p(t) \propto^{EOS} z(t) \tag{2}$$

For complex shaped propellants it is very difficult if not impossible to provide an analytical form function. Even for standard propellant grain shapes like a 7-hole or 19-hole geometry it is very cumbersome to calculate φ(z) especially after the sliver point. For new kinds of propellants like consolidated charges or foamed propellants it is impossible to calculate a form function from the basic geometry of the solid propellant. Therefore ICCA was developed at the Fraunhofer Institute for Chemical Technology. But due to restrictions in computational power, only qualitative simulations have been possible in the past [3][4]. With the emergence of new and more powerful computer systems using new techniques these limitations can be stretched to make quantitative simulations possible.

Basics of the ICT-Cellular-Combustion-Algorithm (ICCA)

The history of cellular automata dates back to the 1940´s when they were first developed by Stanislaw Ulam at the Los Alamos National Laboratory to describe crystal growth and later advanced by work of John von Neumann to study the behavior of biological systems [5]. Since then they have been used in a plethora of applications to model discrete dynamic systems.

A cellular automata is completely defined by a cellular space Z, a set of states Q, a neighborhood N and a local evolution function δ.

The cellular space Z as realized in ICCA consists of a lattice of cells that are pixels in two dimensions and voxel in three dimensions. The cells are initially in a binary state. That

means their state is either 0 or 1. 0 means the cell is empty and 1 means the cell is filled with solid propellant. For two dimensions this is highlighted in Figure 1 and Figure 2 where empty cells are depicted white and cells filled with propellant are black. An individual cell can be indexed by its position in the x-y plane in two dimensions or the x-y-z-position in three dimensions, the notation for the state of the cell at position (x,y) in the i-th step is then $m^{(i)}_{xy}$ the whole array is called the mass-array $m^{(i)}$.

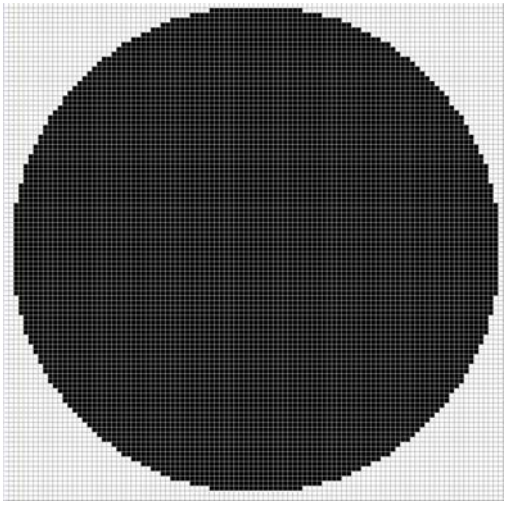

Figure 1: Example for the cellular space Z in ICCA

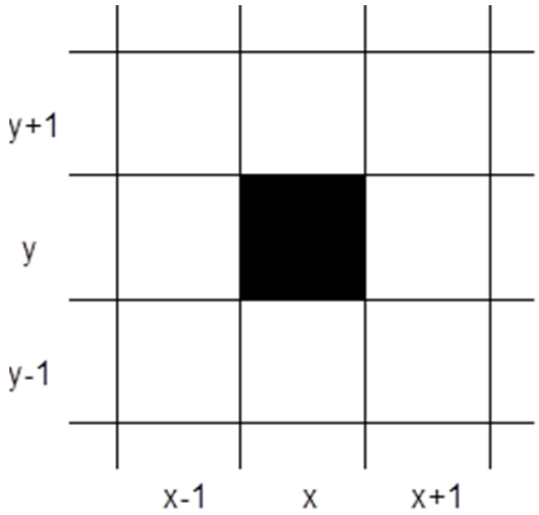

Figure 2: Example for the cellular space in ICCA - close up

The set of states that an individual cell can occupy are the real numbers in the interval between 0 and 1 that is

$$Q = \mathbb{R} \in [0,1] \tag{3}$$

The neighborhood N can be defined in many ways, depending on how many and which neighboring cells are supposed to influence the evolution of a particular cell. In ICCA the neighborhood N is assumed to be a Moore neighborhood. This means that all 8 neighboring cells respectively 26 in three dimensions influence the evolution of a cell. An example for the neighborhood in two dimensions is depicted in Figure 3.

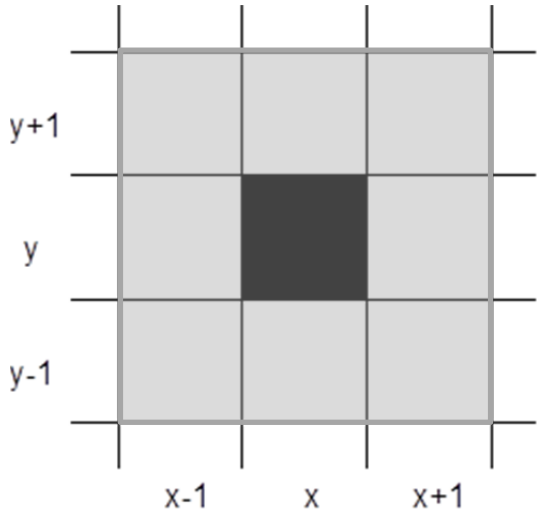

Figure 3: Definition of the neighborhood N in ICCA

For the evolution function δ the following ansatz is used in two dimensions:

$$\Delta_{2d}^{(i)} m_{xy} = \Delta m_0 \cdot {a_{xy}}^{(i-1)} \left({m_{xy}}^{(i-1)}\right)^{\frac{1}{2}} \quad \in [0, \Delta m_0] \tag{4}$$

$\Delta_{2d}^{(i)} m_{xy}$ is the mass reduction for the cell at position (x,y) in the i-th step denoted by $m^{(i)}_{xy}$. $\Delta m_0 \in [0,1]$ is the maximal mass reduction per step, $a_{xy} \in [0,1]$ is a so called activation factor that encodes the influence of the neighborhood N on the state of the cell in the i-th step and $\left(m_{xy}^{(i-1)}\right)^{\frac{1}{2}} \in [0,1]$ is a factor that is proportional to the surface of an individual cell. A generalization to three and more dimensions is straight forward.

Creation of input data

The algorithm in the simple form as described above needs a two dimensional array of 0´s and 1´s, where the location of the 1´s describes the geometry of the solid propellant. This information can be extracted from all kinds of data. The simplest way is to use a black and white bitmap where the black pixels represent filled cells with m_{xy} = 1 as shown in Figure 1. Such a bitmap can be created with standard software in a matter of seconds.

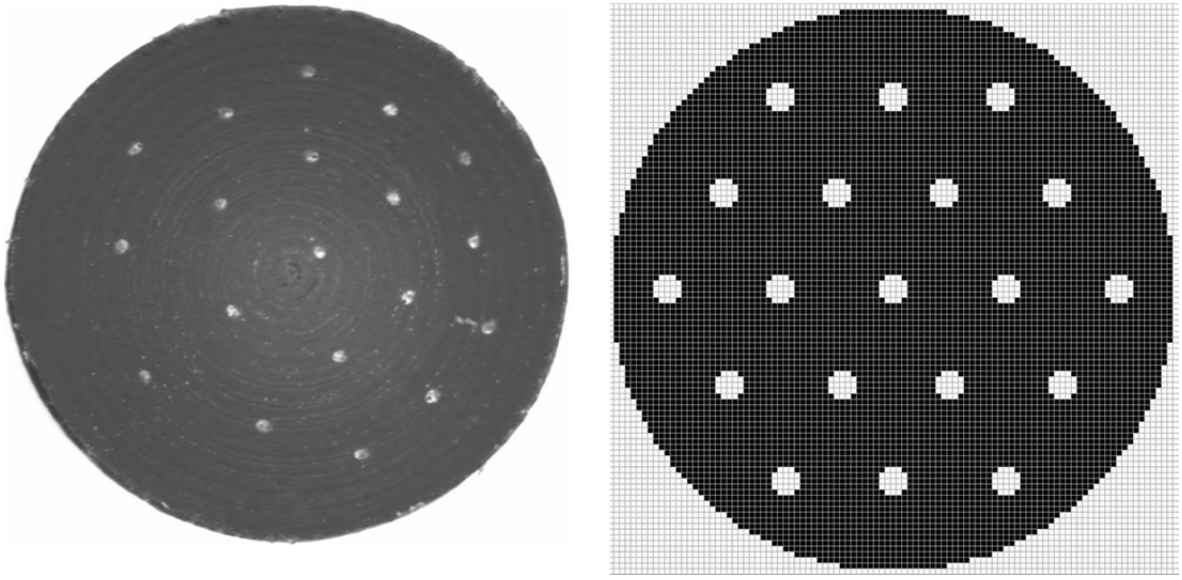

Figure 4: Schematic creation of two dimensional input data for ICCA from photographs

But other ways of creating input data are possible too, e.g. photographs as depicted in Figure 4 or CT-data to generate the necessary input. Especially in three dimensions CT-data represent an easy way to get geometry data of real solid propellant grains.

Connection with the simulation of propellant combustion and visualization of combustion process

To use the algorithm for the simulation of pressure buildup for example in a closed vessel, information about the burning surface and the burned mass fraction have to be extracted from the mass array $m^{(i)}$. Both can be deducted from the state of the array $m^{(i)}_{xy}$ by summation over all cells. The burned mass fraction is simply calculated with a sum of all the completely or partially filled cells (eq. 5).

$$z^{(i)} = 1 - \frac{\sum_{xy} m^{(i)}{}_{xy}}{\sum_{xy} m^{(0)}{}_{xy}} \tag{5}$$

The burning surface can be derived in a similar manner by a summation procedure over the partially burned cells. As a result the form function φ(z) in a tabular form is obtained. φ(z) can now be used together with a burning law (eq. 1) and an equation of state (eq. 2) to simulate the pressure build up or dynamic vivacity e.g. in a closed vessel.

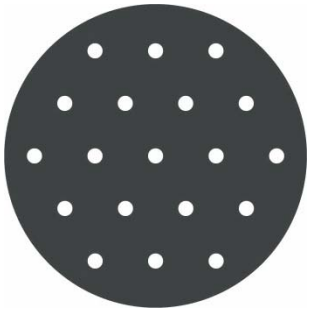

Figure 5: 19-hole grain

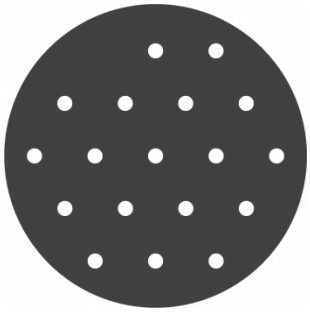

Figure 6: 18-hole grain

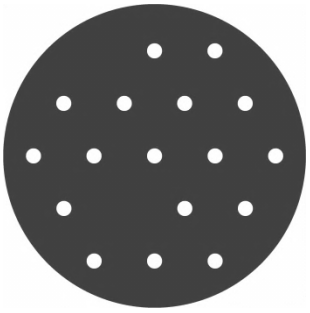

Figure 7: 17-hole grain

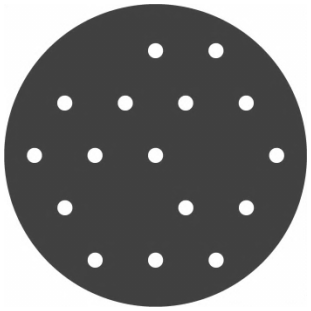

Figure 8: 16-hole grain, variant1

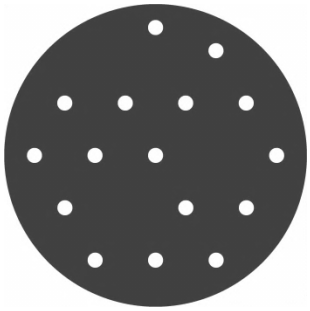

Figure 9: 16-hole grain, variant2

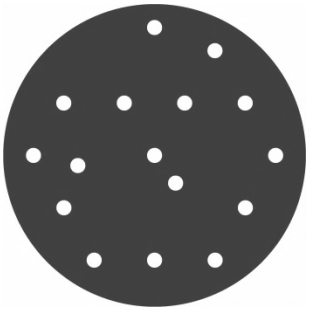

Figure 10: 16-hole grain, variant3

To demonstrate the whole process the input data for a 19-hole propellant grain with diameter 2 mm, a web of 0.3 mm and a hole diameter of 0.1 mm was prepared, see Figure 5. The resolution was chosen to be 1000 Pixel/mm. With this geometry as a starting point the number and position of holes was varied. Prepared input-data with the geometry variations are shown in Figure 6 to Figure 10.

It should be noted that the size of the web was not optimized even for the 19-hole propellant grain, meaning that in all cases there are multiple sliver points present. The resulting form functions are shown in Figure 11.

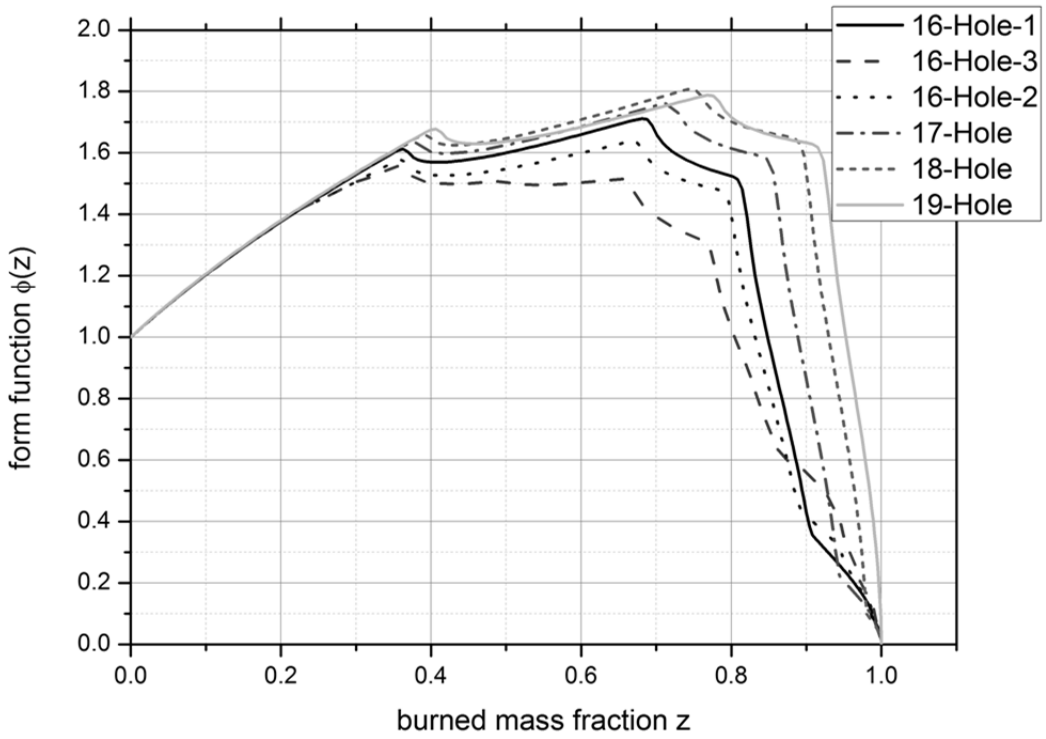

Figure 11: Resulting form functions for shown input, numbers denote 16-hole input variants

As can be clearly seen in Figure 11 at the first sliver point the outer web and at the second one the inner web burns out. The additional steps in the form functions for the 16-hole grains originate in the burning of the web between individual holes due to web variations.

An exemplary pressure simulation without heat-loss is shown in Figure 12. Without going too much into detail about the chosen parameters for the simulation this example is only meant to show that geometry variations and their consequences can be studied in a fast and easy way using the ICT-Cellular-Combustion-Algorithm.

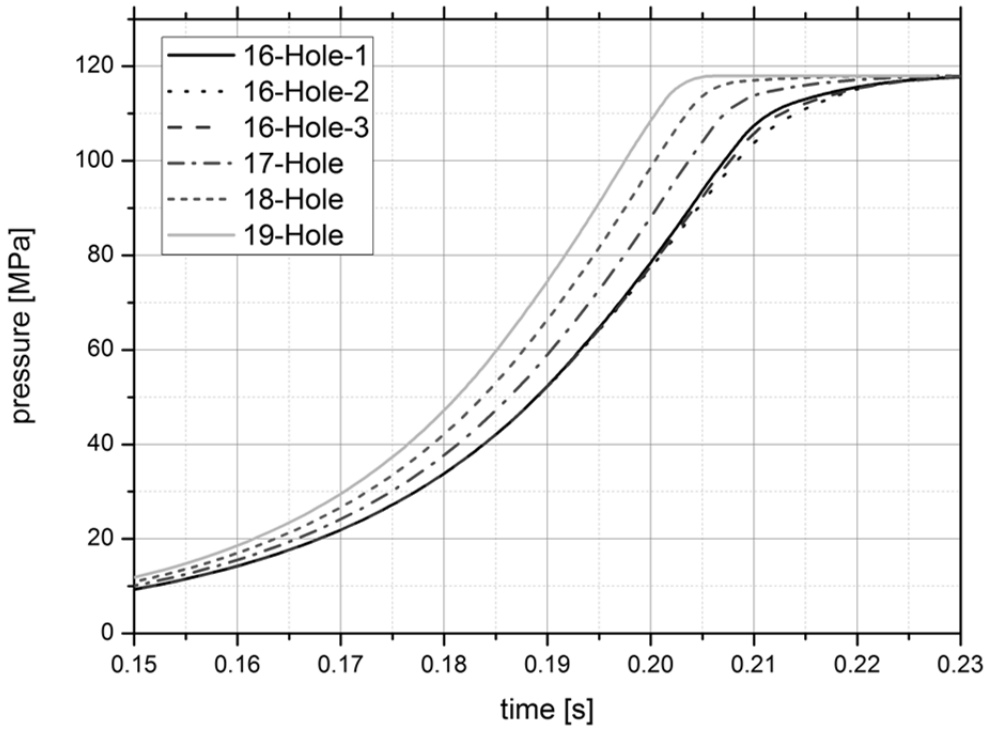

Figure 12: Exemplary pressure simulation for chosen geometries

Another key feature is the ability of the algorithm to provide a visualization of the geometry changes during the combustion process. The only necessary thing to do is to graphically represent the array $m^{(i)}_{xy}$ e.g. in a bitmap. The complete combustion process can also be animated by compiling the sequence of arrays $m^{(i)}_{xy}$ from start to finish in a movie sequence. An example for the visualization is shown in Figure 13, where the 19-hole propellant grain from Figure 5 is shown during combustion after the first sliver point but shortly before the second.

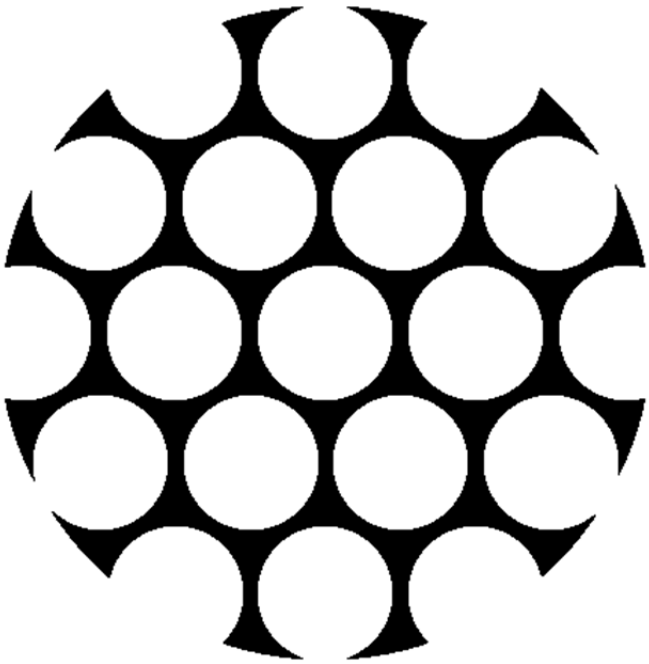

Figure 13: Visualization of 19-hole propellant during combustion

Challenges for the application to the combustion of complex shaped propellants

The quality of the calculated result is directly dependent on the chosen resolution. Simply put if a circle is approximated by only one pixel, the algorithm will deliver only a very crude approximation of the form function. The more pixels are used to approximate a given geometry the better are the results that ICCA delivers. So the need for high resolution calculations is obvious, especially to simulate the combustion of complex shaped propellants.

But if the resolution is increased too much, the limits of computation time are reached very quickly (occupied memory can also be problematic but the computation time usually gets unbearable before the computer runs out of memory). This problem gets even worse in the case of quantitative simulations in three dimensions. This is because the number of calculations per step scales with the number of cells $N_c^{(i)}$ in the mass array. If there are n-cells in each direction x, y and z the number of calculation scales with the cube of n.

$$N_c^{(i)} \propto n^3 \tag{6}$$

There are several possible ways to make high resolution calculations possible and mitigate the effects of this “curse of dimensionality”.

The two strategies that have shown potential to improve the speed of a calculation the most, are dynamic resizing of the array during combustion and parallelization of the algorithm.

Dynamic resizing means that the boundaries of the array $m^{(i)}_{xy}$ shrink during the combustion when the filled cells near the boundaries are already burned up. This is schematically shown in Figure 14.

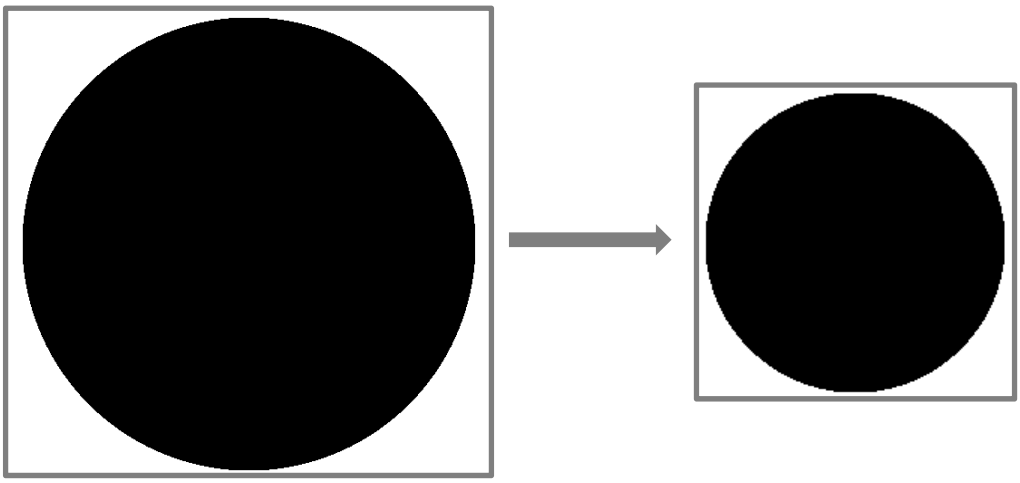

Figure 14: Illustration of dynamic resizing of mass array boundaries, grey frame illustrating mass array limits

The other strategy is based on the ability of modern computer systems to utilize more than one processor-core at once. The Compute Unified Device Architecture (CUDA) created by NVIDIA [6] for example is a programming platform that allows the developer to directly access the graphics processing unit (GPU) in a graphics card. Modern graphics cards usually have hundreds of processing-cores intended for the processing of modern 3d-graphics. Meaning that instead of one operation at a time as with a single core system now hundreds of operations can be carried out simultaneously. Initially intended for computer graphics, CUDA now finds more and more applications in the fields of science where massive computational power is needed.

For ICCA this means that one is able to calculate the evolution of hundreds of cells per step i simultaneously (the exact number depends on the model of the graphics card used). This is possible because the state evolution of a cell in step i depends only on the state of the cells in its neighborhood N at step i-1. This means the evolution of every cell in the mass-array $m^{(i)}$ can be calculated at the same time.

Parallelization and dynamic resizing have been implemented in a feasibility study where a speed up of several orders of magnitude was possible. The combination of both techniques promises to make quantitative simulation of arbitrary shaped solid propellants possible in the

future. A presentation of the work done to speed up the simulations will be given in a separate paper.

Summary

The ICT-Cellular-Combustion-Algorithm (ICCA), a novel algorithm for the calculation of geometrical changes during the combustion of solid propellants, has been presented. The theoretical background has been described and the possibilities to create input data have been demonstrated.

For several different propellant grains the process of creating input data, calculating the form function and simulating the pressure build up in a closed vessel has been demonstrated. It has been shown that ICCA represents an easy and fast way to study geometry changes in standard solid propellants. Another key feature of ICCA is the possibility to provide an easy way for the visualization of the geometry changes during the combustion process.

At last the application of ICCA to the simulation of complex shaped solid propellant combustion was discussed. For complex shaped propellants high resolution calculations are necessary to achieve good results. This leads to unbearably large computation times on single-core computer systems. This is the reason why in the past only qualitative simulations of complex propellants were possible. Two possible solutions to this challenge have been discussed. Dynamic resizing of the mass array during the combustion process on the one hand reduces the number of cells that have to be evolved per calculation step, discarding empty already burned up cells. Parallelization of the algorithm on the other hand making use of the possibilities of modern computer systems to utilize more than one processor-core, for example using CUDA to access the hundreds of processor-cores on a modern graphics card, allows calculating the evolution of more than one cell at a time.

A combination of both techniques promises to make the quantitative simulation of arbitrary shaped solid propellant combustion possible in the future.

Acknowledgment

This work is supported by the Bundeswehr Technical Center for Weapons and Ammunition (WTD 91) in Meppen.

References

[1] Anderson, R.D.; Fickie, K.D.; IBHVG2 – A User´s Guide, BRL-TR-2829, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, July 1987

[2] Volk, F.; Bathelt, H.; User´s Manual for the ICT-Thermodynamic Code; Vol. 2; Proceedings of the 27th International Annual Conference of ICT, Karlsruhe, Germany, 92/1-16, 1996

[3] Fischer, T.S., Meßmer, A.; Langer, G.; Weindl, M.; Modellierung von Abbrandphänomenen bei porösen Ladungen, Proceedings oft the 30th International Annual Conference oft ICT, Karlsruhe, Germany, 98/1-13, 1999

[4] Fischer, T.S.; Messmer, A.; Burning Characteristics of Foamed Polymer Bonded Propellants, 19th International Symposium on Ballistics, May 2001, Interlaken, Switzerland

[5] Von Neumann, J.; The General and Logical Theory of Automata, Cerebral Mechanisms in Behavior: The Hixon Symposium, New York: John Wiley & Sons

[6] Nickolls, J.; Buck, I.; Garland, M.; Skadron, K.; Scalable Parallel Programming with CUDA. *Queue* 6, 2 (March 2008), 40-53. DOI=10.1145/1365490.1365500 http://doi.acm.org/10.1145/1365490.1365500