

Problem-Solving Environment for Beam Dynamics Analysis in Particle Accelerators

Nataliia Kulabukhova^(✉), Alexander Bogdanov, and Alexander Degtyarev

Saint-Petersburg State University, St. Petersburg, Russia
n.kulabukhova@spbu.ru

Abstract. In particle accelerator physics the problem is that we can not see what is going on inside the working machine. There are a lot of packages for modelling the behaviour of the particles in numerical or analytical way. But for most physicists it is better to see the picture in motion to say exactly what is happening and how to influence on this. The goal of this work is to provide scientists with such a problem-solving environment, which can not only do some numerical calculations, but show the dynamics of changes as a motion 3D picture. To do this we use the power of graphical processors from both sides: for general purpose calculations and for there direct appointment – drawing 3D motion. Besides, this environment should analyse the behaviour of the system to provide the user with all necessary information about the problem and how to deal with it.

1 Introduction

Today accelerator physics is a multidisciplinary kind of science. Not only physicists working in this area. Mathematicians, engineers, programmers are also involved in this process. It is obvious that the progress of further research depends more on new technologies and IT industry, which can be truly helpful for theoretical physics. Of course, all this can be said not only about accelerator physics [1, 2].

There are a lot of well known software packages, which are used practically everywhere. Our opinion is that the use of this or that package depends mostly on the laboratory and the research. However, the main are COSY Infinity [3], MaryLie [4], MAD [5], Trace (with different modifications) [6]. Some of them are cross-platform, others not. Though there are a FORTRAN-based applications [7, 8] among them, there trustful and stable work is out of doubt. That is the reason why scientists do not want to change anything and start working with other applications. To tell the truth, there is no new modern product that is really different from the old ones. COSY Infinity is working in the field of nonlinear dynamics, where it is used for the computation of perturbation expansions of Poincare maps to high orders. In MaryLia another approach is used. It is based on Lie algebraic methods, which may be used for particle tracking

The work is supported by RFBR 16-07-01113A.

around or through a lattice and for analysis of linear and nonlinear lattice properties. In addition to single-particle tracking, Lie algebraic methods may also be used to determine how particle phase-space distribution functions evolve under transport through both linear and nonlinear elements. These methods are useful for the self-consistent treatment of space-charge effects and for the study of how moments and emittance evolve. These two approaches are rather quick, but the disadvantage is the need of recalculate all the equations if it is necessary to change the parameter.

Alternative to described above approaches became the Particle-In-cell method [9–11], which came from gas dynamics at the beginning of the 21 century. The main advantage of PIC method is that it does not need to approximate the equation of state. On the other hand, it is very much time depending, because of the amount of data. At present, there are a lot of modifications of PIC method using parallelization on multiple processors [12], hybrid systems [13] and the Intel XEON PHI [14]. But it does not crucially influence on the problem of data sending.

The other idea of helping scientists with there work is to solve the problem of analysing data. There are plenty of works [15–19], where the words “Virtual Accelerator” standing near by with the control system describing. The main task for such environment is to collect data from different detectors and handled them before it can be analysed. On the Fig. 1 the work of such system is shown. The machine operator is the essential component of it. The work of operator is to coordinate the send-received collected data between different detectors and

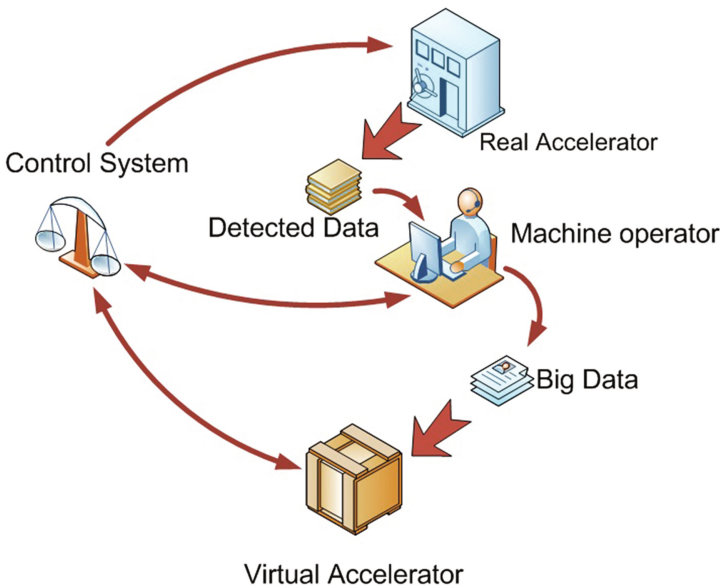


Fig. 1. Working process of virtual accelerator system

the Virtual Accelerator system. Besides, the operator detects the control system to get the instructions for what should be done to fix the errors and improve the results. But all this Virtual Accelerators are just program module working above the real machine. They are rigidly tied to one type of accelerators, and need crucial changes to be used on another.

However, the software packages for modelling beam dynamics and Virtual Accelerator for analysing data are without a doubt an essential part of accelerator physics. On our point of view, it is better to merge these two approaches and make them independent from real facilities.

2 The Main Concept

The idea of using 3D graphics for visualization is a part of a huge project of constructing so called Virtual Accelerator Laboratory (VAL) [20,21]. The idea is to develop a model of real machine and test the behaviour of the beam inside this virtual system of control elements. As it was said above, there are a lot of projects using the words of “Virtual Accelerator” [22] and a number of packages to modelling beam dynamics. The concept of VAL is to provide the user (scientist) with block of different components, such as:

the block of control elements: with the help of it the user can construct the main view of the future accelerator. The general view of circular machine is shown on Fig. 2.

the block of particle distribution: which forms the initial particle distribution coming from the source.

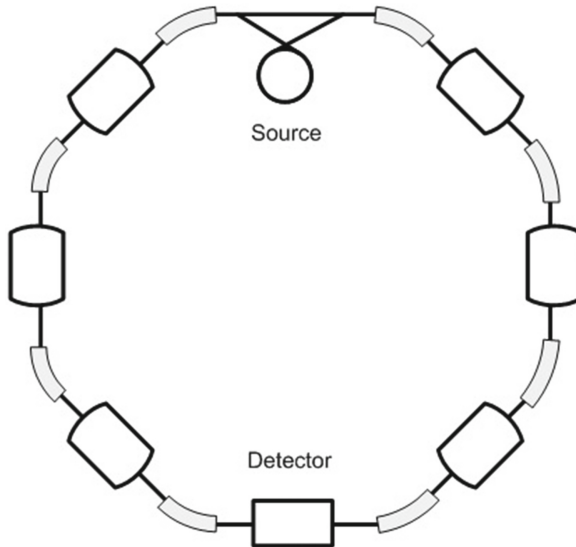


Fig. 2. General view of accelerator ring

the block of data: the data base with all calculated results.

the block of expert system: where the collected data is analysed and the main reasons of received errors are given as a result to user (see Fig. 3).

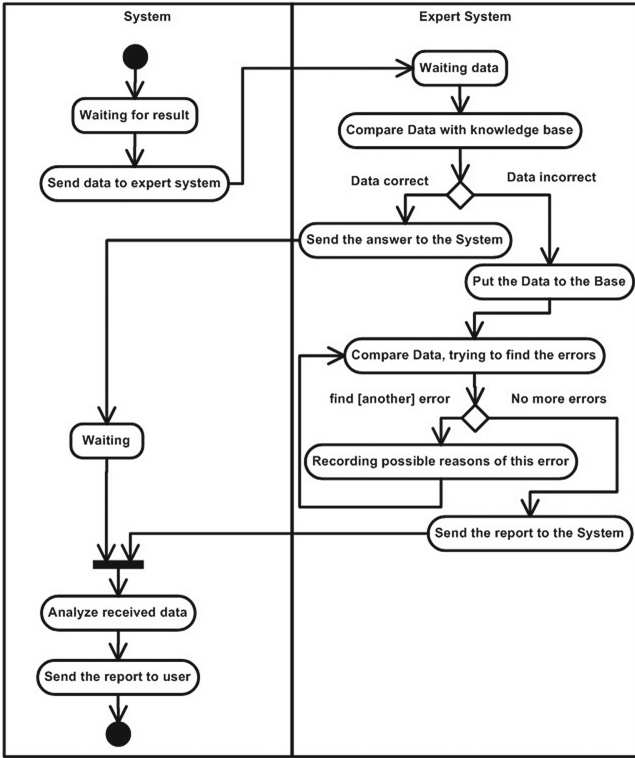


Fig. 3. The working process of the expert system

The advantage of using such VAL is that we can construct a machine before it is really exist. And based on the results of the expert system the advices to build the real accelerator can be done. That significantly saves time and financial resources. Besides, the prediction of particle motion with different nonlinear effects can be made. VAL works in a distributed environment that is why, the user of the system can use different electronic devices to check the intermediate results of the work. Figure 4 shows this idea.

On Fig. 5 the general scheme of program modules is shown.

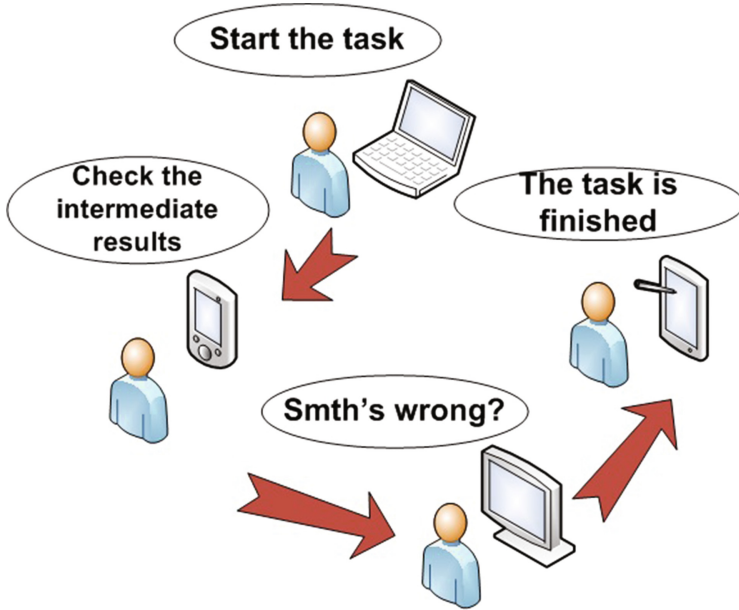


Fig. 4. Distributed environment of VAL

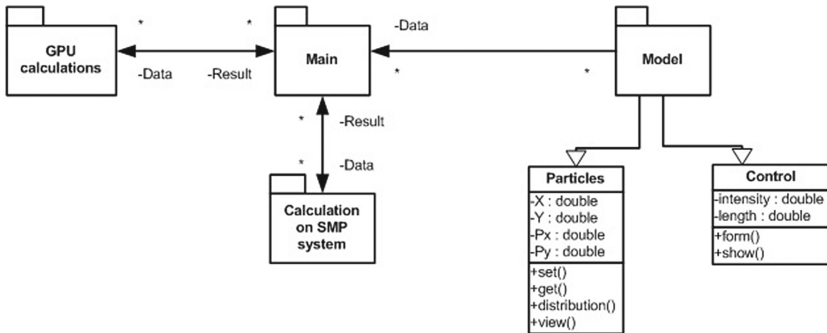


Fig. 5. Main modules of the system

3 About the Approach

The main idea of the method was described in [20]. We have the initial distribution and the control system of elements. If we speak about elements, everything is well known and there are no great problems to construct such system. On the contrary, the difficulty is how to form the initial distribution. There is no proved and stable approach to do this easily.

The easy task is to work with the uniform distribution, but in real case we could not get it from the source. We are trying to use normal and some

other distributions, described in [23], to make it more natural. Passing through the control system the initial distribution, created on the first step and formed as matrix is modified in new intermediate matrix \mathbf{X} .

$$\mathbf{X}_0 = \begin{pmatrix} X_1 & X_2 & \dots & X_n \\ Y_1 & XY_2 & \dots & Y_n \\ Px_1 & Px_2 & \dots & Px_n \\ Py_1 & Py_2 & \dots & Py_n \\ \dots & \dots & \dots & \dots \end{pmatrix}$$

Following the predictor-corrector algorithm [24] this new matrix is the base matrix to correct the initial distribution and form it for the new turn. The Fig. 6 shows this idea.

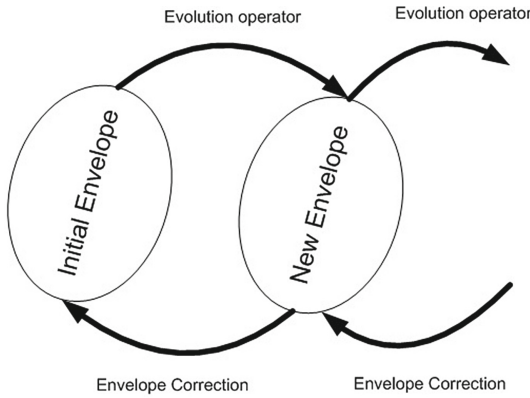


Fig. 6. Correction of beam distribution

The general view of the equation of motion is

$$X = \sum_{i=0}^k P^{1i}(t)X_0^{[i]} \tag{1}$$

In the case of space charge, using the described approach [20], Eq. (1) can be rewritten

$$\frac{d\mathbf{X}}{ds} = \sum_{k=1}^{\infty} (\mathbb{P}_{ext}^{1k}(s) + \mathbb{P}_{self}^{1k}(s)) \mathbf{X}^{[k]} \tag{2}$$

For example, the control matrix \mathbb{P}^{11} in this case can take the following view

$$\mathbb{P}^{11}(s) = \mathbb{P}_{ext}^{11}(s) + \mathbb{P}_{self}^{11}(s)$$

where

$$\mathbb{P}_{ext}^{11}(s) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -k_x & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -k_y & 0 \end{pmatrix},$$

$$\mathbb{P}_{self}^{11}(s) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ -\eta_x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \eta_y & 0 \end{pmatrix},$$

If we speak about nonlinear case, the described above equations can be also used with some modifications.

The matrix for \mathbf{X} is formed by the Kronecker composition:

$$\mathbf{X}^3 = \begin{pmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{X}^{[3]} \\ \mathbf{X}^{[2]} \otimes \mathbf{Y} \\ \mathbf{X} \otimes \mathbf{Y}^{[2]} \\ \mathbf{Y}^{[3]} \end{pmatrix}.$$

And for matrices \mathbb{P} we add new order:

$$\mathbb{P}^{11} = \begin{pmatrix} \mathbb{P}_x^{11} & \mathbb{O} \\ \mathbb{O} & \mathbb{P}_y^{11} \end{pmatrix},$$

$$\mathbb{P}^{13} = \begin{pmatrix} \mathbb{Q}_x^{11} & \mathbb{O} & \mathbb{Q}_x^{13} & \mathbb{O} \\ \mathbb{O} & \mathbb{Q}_y^{22} & \mathbb{O} & \mathbb{Q}_y^{24} \end{pmatrix},$$

where, for example, elements of \mathbb{Q}_x^{11} can be

$$\mathbb{A}(1, 1) = h \frac{13 \cos(l\sqrt{|h|}) - \cos(3l\sqrt{|h|}) - 36l\sqrt{|h|} \sin(l\sqrt{|h|})}{192}$$

$$\mathbb{A}(1, 2) = \sqrt{|h|} \frac{-5 \sin(l\sqrt{|h|}) - 13 \sin(3l\sqrt{|h|}) + 12l\sqrt{|h|} \cos(l\sqrt{|h|})}{64}$$

$$\mathbb{A}(1, 3) = - \frac{\cos(l\sqrt{|h|}) - \cos(3l\sqrt{|h|}) + 12l\sqrt{|h|} \sin(l\sqrt{|h|})}{64}$$

$$\mathbb{A}(1, 4) = \frac{36l\sqrt{|h|} \cos(l\sqrt{|h|}) + 13 \sin(3l\sqrt{|h|}) - 75 \sin(l\sqrt{|h|})}{192\sqrt{|h|}}$$

4 Reasons of Using Graphical Processors

As it was said above there are a lot of works describing the parallelization of principally PIC method (see works [12,25]). Though, the approach we use is a matrix based method of Ordinary Differential Equation solving and it gives us some advantages of paralleling them on SMP systems, for example, it became clear that there it is no matter how much resources we have, using only this technologies we a limited. As the method allows to present the intermediate results and the solution of the system in the form of matrices it is better to use the graphical processors [26].

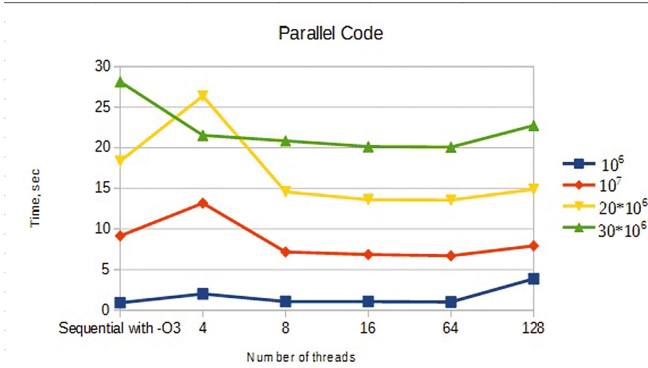


Fig. 7. Parallel code different number of turns using OpenMP concept

The present research shows that there is no great benefit via parallelization of computational code for one particle by using OpenMP library (see Fig. 7). In this case overhead on data sending is significant. On the other hand, we forms the initial distribution and the intermediate results in the form of matrices and that allows us to use GPUs for there direct appointment. But the idea is to use graphical processor not only for general purpose but for drawing as well.

To simplify this process the 3D Virtual Cluster [27] is going to be used. It helps to visualized the results of calculations with some modified based resources.

5 Conclusion

Our goal is to provide the scientist with a clear and easy environment to simulate and predict the behaviour of the beam. The result of the simulation should be shown as a motion picture. We do not guarantee the work of the system in the real time, but it should be the problem-solving environment, that can help the user to construct the needed structure of the accelerator. The previous work was devoted to the particle simulation using matrix approach. Some parallelization of this method was made. But the results showed that there can be some other benefits for modelling beam dynamics using this approach. In this work we sum up all that was done before to transform the simulation method to the problem-solving environment.

Acknowledgements. The authors would like to express gratitude to Vladimir Korkhov for valuable help. And Serge Andrianov for useful explanations of physical processes in accelerators. Scientific research were performed using the equipment of the Research Park of St. Petersburg State University. The work was sponsored by the Russian Foundation for Basic Research under the projects: 16-07-01113 “Virtual supercomputer as a tool for solving complex problems”.

References

1. Raba, N.O., Stankova, E.N.: On the problem of numerical modeling of dangerous convective phenomena: possibilities of real-time forecast with the help of multi-core processors. In: Murgante, B., Gervasi, O., Iglesias, A., Taniar, D., Apduhan, B.O. (eds.) ICCSA 2011. LNCS, vol. 6786, pp. 633–642. Springer, Heidelberg (2011). doi:[10.1007/978-3-642-21934-4_51](https://doi.org/10.1007/978-3-642-21934-4_51)
2. Stankova, E.N., Korkhov, V.V., Kulabukhova, N.V., Vasilenko, A.Y., Holod, I.I.: Computational environment for numerical modeling of the results of cloud seeding. In: Gervasi, O., Murgante, B., Misra, S., Rocha, A.M.A.C., Torre, C., Taniar, D., Apduhan, B.O., Stankova, E., Wang, S. (eds.) ICCSA 2016. LNCS, vol. 9788, pp. 454–462. Springer, Cham (2016). doi:[10.1007/978-3-319-42111-7_35](https://doi.org/10.1007/978-3-319-42111-7_35)
3. Makino, K., Berz, M.: COSY INFINITY Version 9. Nuclear Instruments and Methods A558 (2005)
4. Dragt, A.J., Ryne, R.D., et al.: Numerical computation of transfer maps using lie algebraic methods. In: Proceedings of PAC 1987 (1987)
5. Ryne, R.D.: Advanced computing tools and models for accelerator physics. In: Proceedings of EPAC 2008 (2008)
6. <http://laag.lanl.gov/laag/services/traceman.pdf>
7. <http://cosyinfinity.org/>
8. Dragt, A.J., Ryne, R.D., et al.: MARYLIE 3.0 users manual: a program for charged particle beam transport based on lie algebraic methods. University of Maryland (2003)
9. Paret, S., Qiang, J.: Collisional effects in particle-in-cell beam-beam simulation. In: Proceedings of IPAC 2013. JACOW (2013)
10. Wolfheimer, F., Gjonaj, E., Weiland, T.: Parallel Particle-In-Cell (PIC) codes. In: Proceedings of ICAP 2006. JACOW (2006)
11. Stancari, G., Redaelli, S., Moens, V.: Beam dynamics in an electron lens with the warp Particle-In-Cell code. In: Proceedings of IPAC 2014. JACOW (2014)
12. Giovannozzi, M.: Space-Charge Simulation Using Parallel Algorithms
13. Bowers, K.J.: Accelerating a Particle-in-Cell simulation using a hybrid counting sort. *J. Comput. Phys.* **173**, 393–411 (2001)
14. Meerov, I.B., et al.: 3D plasma modelling PIC method on Intel XEON PHI: optimisation and Examples of Use (in Russian), vol. 15. Computational Methods and Programming (2015)
15. Chiu, N.P.C., Kuo, C.H., Chen, J., Cheng, Y.S., Wu, C.Y., Chen, Y.K., Hsu, K.T.: Virtual accelerator development for the TPS. In: Proceedings of IPAC 2010. JACOW (2010)
16. Gu, D., Zhang, M., Gu, Q., Huang, D., Zhao, M.: Development of virtual accelerator environment for beam diagnostics (2014). <https://arxiv.org/ftp/arxiv/papers/1401/1401.1889.pdf>
17. Gulliford, C., Bazarov, I., Dobbins, J., Talman, R., Malitsky, N.: The NTMAT EPICS-DDS virtual accelerator for the Cornell ERL injector. In: Proceedings of IPAC 2010. JACOW (2010)
18. Malitsky, N., Smith, J., Wei, J., Talman, R.: UAL-based simulation environment for spallation neutron source ring. In: Proceedings of the 1999 Particle Accelerator Conference. JACOW (1999)
19. Sagan, D., et al.: Unified accelerator modeling using the BMAD Software Library. In: Proceedings of IPAC 2011. JACOW (2011)

20. Kulabukhova, N., Andrianov, S.N., Bogdanov, A., Degtyarev, A.: Simulation of space charge dynamics in high intensive beams on hybrid systems. In: Gervasi, O., Murgante, B., Misra, S., Rocha, A.M.A.C., Torre, C., Taniar, D., Apduhan, B.O., Stankova, E., Wang, S. (eds.) ICCSA 2016. LNCS, vol. 9786, pp. 284–295. Springer, Cham (2016). doi:[10.1007/978-3-319-42085-1_22](https://doi.org/10.1007/978-3-319-42085-1_22)
21. Kulabukhova, N.: Software for virtual accelerator environment. In: RuPAC 2012 Contributions to the Proceedings. JACOW (2012)
22. Yamamoto, N.: Use of a virtual accelerator for a development of an accelerator control system (1998). <http://accelconf.web.cern.ch/accelconf/pac97/papers/pdf/3P042.PDF>
23. Nataliia, K.: Space charge dominated envelope dynamics using GPUs. In: Proceedings of IPAC 2013. JACOW (2013)
24. Kulabukhova, N., Degtyatev, A., Bogdanov, A., Andrianov, S.: Simulation of space charge dynamics on HPC. In: Proceedings of IPAC 2014. JACOW (2014)
25. Qiang, J., Ryne, R.D., Habib, S., Decy, V.: An object-oriented parallel particle-in-cell code for beam dynamics simulation in linear accelerators (2000)
26. Nataliia, K.: GPGPU implementation of matrix formalism for beam dynamics simulation. In: Proceedings of ICAP 2012. JACOW (2012)
27. Bogdanov, A., Ivashchenko, A., Belezeko, A., Korkhov, V., Kulabukhova, N., Khmel, D., Suslova, S., Milova, E., Smirnov, K.: Building a virtual cluster for 3D graphics applications. In: Gervasi, O., Murgante, B., Misra, S., Rocha, A.M.A.C., Torre, C., Taniar, D., Apduhan, B.O., Stankova, E., Wang, S. (eds.) ICCSA 2016. LNCS, vol. 9787, pp. 276–291. Springer, Cham (2016). doi:[10.1007/978-3-319-42108-7_21](https://doi.org/10.1007/978-3-319-42108-7_21)