Sio-Iong Ao · Haeng Kon Kim
Mahyar A. Amouzegar *Editors*

Transactions on Engineering Technologies

World Congress on Engineering and Computer Science 2015

Springer

Transactions on Engineering Technologies

Sio-Iong Ao · Haeng Kon Kim
Mahyar A. Amouzegar
Editors

Transactions on Engineering Technologies

World Congress on Engineering and Computer Science 2015

Springer

Editors
Sio-Iong Ao
IAENG Secretariat
International Association of Engineers
Hong Kong
Hong Kong

Mahyar A. Amouzegar
Provost and Senior Vice-President
University of New Orleans
New Orleans, LA
USA

Haeng Kon Kim
Department of Computer and Communication
Engineering College, Catholic University of Daegu
Daegu
South Korea

ISBN 978-981-10-2716-1 ISBN 978-981-10-2717-8 (eBook)
DOI 10.1007/978-981-10-2717-8

Library of Congress Control Number: 2016931995

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #22-06/08 Gateway East, Singapore 189721, Singapore

Preface

A large international conference on Advances in Engineering Technologies and Physical Science was held in San Francisco, California, USA, October 21–23, 2015, under the auspices of the World Congress on Engineering and Computer Science (WCECS 2015). The WCECS 2015 was organized by the International Association of Engineers (IAENG). IAENG, originally founded in 1968, is a non-profit international association for the engineers and the computer scientists. The WCECS Congress serves as an excellent platform for the members of the engineering community to meet and exchange ideas. The congress in its long history has found a right balance between theoretical and application development, which has attracted a diverse group of researchers, leading to its rapid expansion. The conference committees have been formed with over two hundred members including research center heads, deans, department heads/chairs, professors, and research scientists from over 30 countries. The full committee list is available at the congress' web site: www.iaeng.org/WCECS2015/committee.html. WCECS conference is truly an international meeting with a high level of participation from many countries. The response to WCECS 2015 conference call for papers was outstanding, with more than six hundred manuscript submissions. All papers went through a rigorous peer-review process and the overall acceptance rate was 50.89 %.

This volume contains 42 revised and extended research articles, written by prominent researchers, participating in the congress. Topics include engineering mathematics, electrical engineering, communications systems, computer science, chemical engineering, systems engineering, manufacture engineering, and industrial applications. This book offers the state of the art of tremendous advances in engineering technologies and physical science and applications; it also serves as an exceptional source of reference for researchers and graduate students working with/on engineering technologies and physical science and applications.

Hong Kong, Hong Kong — Sio-Iong Ao
Daegu, South Korea — Haeng Kon Kim
New Orleans, LA, USA — Mahyar A. Amouzegar

Contents

Chapter 1
Estimate the Impact of Different Heat Capacity Approximation Methods on the Numerical Results During Computer Simulation of Solidification

Robert Dyja, Elzbieta Gawronska, Andrzej Grosser, Piotr Jeruszka and Norbert Sczygiol

1.1 Introduction

Aluminum alloys are very interesting material widely used in industry. Modeling and computer simulation are one of the most effective methods of studying difficult problems in foundry and metallurgical manufacture. Numerical simulations are used for optimization of casting production. In many cases they are the only possible techniques for carrying out the experiments whose real statement is complicated. Computer modeling allows to define the major factors of a quality estimation of alloy castings. Simulations help to investigate interaction between solidifying casting and changes of its parameters or initial conditions. That process defines the quality of casting, and the problem of adequate modeling of foundry systems. The process mainly depends on the solution of heat equations [1].

Increasing capacity of computer memory makes it possible to consider growing problem sizes. At the same time, increased precision of simulations triggers even greater load. There are several ways to tackle this kind of problems. For instance, one

R. Dyja (✉) · E. Gawronska · A. Grosser · P. Jeruszka · N. Sczygiol
Czestochowa University of Technology, Dabrowskiego 69,
PL42201 Częstochowa, Poland
e-mail: robert.dyja@icis.pcz.pl

E. Gawronska
e-mail: elzbieta.gawronska@icis.pcz.pl

A. Grosser
e-mail: andrzej.grosser@icis.pcz.pl

P. Jeruszka
e-mail: piotr.jeruszka@icis.pcz.pl

N. Sczygiol
e-mail: norbert.sczygiol@icis.pcz.pl

S.-I. Ao et al. (eds.), *Transactions on Engineering Technologies*,
DOI 10.1007/978-981-10-2717-8_1

can use parallel computing [2, 3], someone else may use accelerated architectures such as GPUs [4] or FPGAs [5], while another person can use special organization of computations [6–9].

Solidification may take place at a constant temperature or in the temperature range [10]. If solidification occurs at a constant temperature, it is then referred to as the so-called Stefan problem or the solidification problem with zero temperatures range. Pure metals or alloys of certain specific chemical compositions (e.g. having an eutectic composition) solidify at a constant temperature. However, most of the metal alloys solidify in certain temperature ranges (so-called temperature intervals of solidification). The temperature at which the alloy starts to solidify is called liquidus temperature (T_l), and the temperature at which solidification ends is called solidus temperature (T_s). In the case of alloys with eutectic transformation, in which the solute concentration exceeds its maximum solubility in the solid phase, the temperature of the solidification end is the eutectic temperature. Analytical (rarely) and numerical (commonly) methods are used in the modeling of solidification process. The finite elements method (FEM) is the most commonly used numerical method, but finite difference method (FDM), boundary element method (BEM), the Monte-Carlo and other methods are also used.

The most important heat effect, occurring during solidification, is the emission of (latent) heat of solidification (L). It is also the most difficult phenomenon to numerical modeling. The basic division of numerical methods of solidification modeling process relates to modeling of the latent heat emission. These methods can be divided into front-tracking methods and fixed-grid methods. Fixed-grid methods are also divided into temperature formulations (the latent heat of solidification is considered as the temperature-dependent term of heat source) and enthalpy formulations (the latent heat of solidification is considered as the temperature-dependent term of heat capacity) [11–16]. The enthalpy methods are divided into methods in which the effective heat capacity depends on the temperature and those in which the effective heat capacity depends on the enthalpy. In our article, we have focused on solving the solidification in the temperature range with the finite element method with the use of fixed-grid methods in enthalpy formulation. We have described the comparison of different ways of approximation of heat capacity in apparent heat capacity (AHC) formulation of solidification. This paper is the extention of previous work [17].

During the comparison, we have used Finite Element Method as numerical method of choice. The FEM was used in our own software, that is used to conduct simulations of solidification. This software is implemented in the C++ programming language with the use of essential libraries, i.e. PETSc (for linear algebra) [18] and TalyFEM (for procedures used in FEM) [19].

1.2 Description of the Entalphy Formulation

Solidification is described by a quasi-linear equation of heat conduction, considering a term of heat source $\dot{q}$ as a latent heat of solidification:

$$\nabla \cdot (\lambda \nabla T) + \dot{q} = c\rho \frac{\partial T}{\partial t} \tag{1.1}$$

By entering the following designation:

$$\dot{s} = \dot{q} - c\rho \frac{\partial T}{\partial t} \tag{1.2}$$

equation (1.1) can be written as

$$\nabla \cdot (\lambda \nabla T) + \dot{s} = 0 \tag{1.3}$$

where $\dot{s}$ denotes generalized heat source. By introducing enthalpy, defined as:

$$h = \int_{T_{ref}}^{T} c\rho(T)\, dt \tag{1.4}$$

where T_{ref} is the reference temperature, and calculating the derivative with respect to the temperature:

$$\frac{dH}{dT} = c\rho(T) = c^*(T) \tag{1.5}$$

where c^* is the effective heat capacity. Assuming the heat source is equal to zero, the Eq. (1.3) can be converted to the form:

$$\nabla \cdot (\lambda \nabla T) = c^* \frac{\partial T}{\partial t} \tag{1.6}$$

All above equations form the basis of the thermal description of solidification.

1.2.1 The Enthalpy and the Effective Heat Capacity

The enthalpy is the sum of explicit and latent heat [20]. For the metal solidifying in the temperature range (T_s–T_l) amounts to:

$$H = \int_{T_{ref}}^{T} c\rho_s(T)\, dT, \quad \text{for} \quad T < T_s,$$

$$
\begin{aligned}
H &= \int_{T_{ref}}^{T_s} c\rho_s(T)\, dT + \int_{T_s}^{T} (\rho_s(T)\frac{dL}{dT} + \\
&\quad + c\rho_f(T)\big)\, dT, \quad \text{for} \quad T_s \le T \le T_l, \\
H &= \int_{T_{ref}}^{T_s} c\rho_s(T)\, dT + \\
&\quad + \rho_s(T)L + \int_{T_s}^{T_l} c\rho_f(T)\, dT + \\
&\quad + \int_{T_l}^{T} c\rho_l(T)\, dT, \quad \text{for} \quad T > T_l
\end{aligned}
\tag{1.7}
$$

The integration of the expressions in Eq. 1.7 gives

$$
\begin{aligned}
c^* &= c\rho_s, \quad \text{for} \quad T < T_s, \\
c^* &= c\rho_f + \rho_s \frac{dL}{dT}, \quad \text{for} \quad T_s \le T \le T_l, \\
c^* &= c\rho_l, \quad \text{for} \quad T > T_l.
\end{aligned}
\tag{1.8}
$$

Assuming that the heat of solidification is exuded and spread evenly throughout the temperature range of solidification, the following can be written:

$$
\begin{aligned}
c^* &= c\rho_s, \quad \text{for} \quad T < T_s, \\
c^* &= c\rho_f + \rho_s \frac{L}{T_l - T_s}, \quad \text{for} \quad T_s \le T \le T_l, \\
c^* &= c\rho_l, \quad \text{for} \quad T > T_l.
\end{aligned}
\tag{1.9}
$$

On the basis of the Eq. 1.7 and the Eq. 1.9 one can make the following graphical comparison of the enthalpy and the effective thermal capacity distributions for alloy solidifying in the temperature range (see Fig. 1.1).

1.2.2 The Types of the Entalphy Formulations

There are three types of enthalpy formulations of solidification:

- basic enthalpy formulation (BEF)

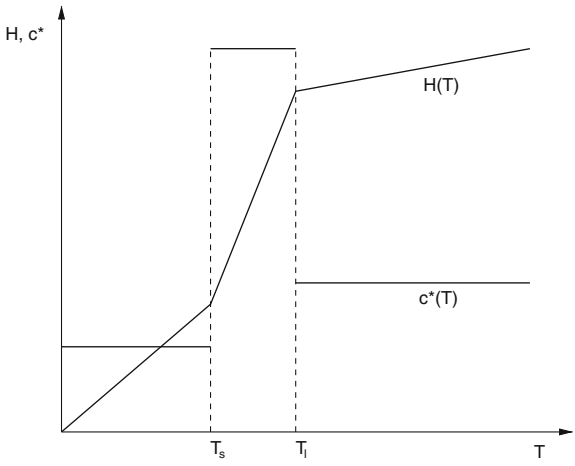

Fig. 1.1 Distribution of enthalpy and effective heat capacity depending on temperature

$$\nabla \cdot (\lambda \nabla T) = \frac{\partial H}{\partial t} \tag{1.10}$$

where

$$H(T) = \int_{T_{ref}}^{T} c\rho \, dT + (1 - f_s(T))\rho_s L \tag{1.11}$$

- apparent (or modified) heat capacity (AHC) formulation
  differentiate Eq. 1.11 with respect to temperature is obtained

$$\frac{dH}{dT} = c\rho - \rho_s L \frac{df_s}{dT} = c^*(T) \tag{1.12}$$

Since $H = H(T(x,t))$ then

$$\frac{\partial H}{\partial t} = \frac{dH}{dT}\frac{\partial T}{\partial t} = c^*(T)\frac{\partial T}{\partial t} \tag{1.13}$$

Substituting Eq. 1.13 to Eq. 1.10 is obtained

$$\nabla \cdot (\lambda \nabla T) = c^*(T)\frac{\partial T}{\partial t} \tag{1.14}$$

- source term formulation (STF)
  The total enthalpy is divided into two parts in accordance with:

$$H(T) = h(T) + (1 - f_s)\rho_s L \tag{1.15}$$

where

$$h(T) = \int_{T_{ref}}^{T} c\rho \, dT \tag{1.16}$$

Derivative Eq. 1.15 with respect to time is

$$\frac{\partial H}{\partial t} = \frac{\partial h}{\partial t} - \rho_s L \frac{\partial f_s}{\partial t} \tag{1.17}$$

Substituting Eq. 1.17 to Eq. 1.10 is obtained

$$\nabla \cdot (\lambda \nabla T) + \rho_s L \frac{\partial f_s}{\partial t} = \frac{\partial h}{\partial t} \tag{1.18}$$

1.3 Approximation of the Effective Heat Capacity

The effective heat capacity can be also calculated directly from the Eq. 1.5, but in this paper, we have presented the results of solidification simulations using the various methods of effective heat capacity approximation.

1. Morgan method—derivative of enthalpy is replaced by a backward differential quotient

$$c^* = \frac{H^n - H^{n-1}}{T^n - T^{n-1}} \tag{1.19}$$

   where $n-1$ and n are the time levels. In some cases, however, this substitution may lead to oscillations in the solution, especially near the boundaries of the temperature range of solidification.
2. Del Giudice method—in order to remove oscillations one should take into account the directional cosines of temperature gradient

$$c^* = \frac{\dfrac{\partial H}{\partial n}}{\dfrac{\partial T}{\partial n}} = \frac{\dfrac{\partial H}{\partial x_i}\alpha_{ni}}{\dfrac{\partial T}{\partial n}}$$

   where

$$\alpha_{ni} = \frac{\dfrac{\partial T}{\partial x_i}}{\dfrac{\partial T}{\partial n}}$$

and

$$\frac{\partial T}{\partial n} = \left(\frac{\partial T}{\partial n} \cdot \frac{\partial T}{\partial n}\right)^{\frac{1}{2}}$$

Hence

$$c^* = \frac{\frac{\partial H}{\partial x}\frac{\partial T}{\partial x} + \frac{\partial H}{\partial y}\frac{\partial T}{\partial y} + \frac{\partial H}{\partial z}\frac{\partial T}{\partial z}}{(\frac{\partial T}{\partial x})^2 + (\frac{\partial T}{\partial y})^2 + (\frac{\partial T}{\partial z})^2} = \frac{H_{,i}T_{,i}}{T_{,j}T_{,j}} \tag{1.20}$$

3. Lemmon method—the temperature gradient is normal to solidification surface

$$c^* = \sqrt{\frac{\left(\frac{\partial H}{\partial x}\right)^2 + \left(\frac{\partial H}{\partial y}\right)^2 + \left(\frac{\partial H}{\partial z}\right)^2}{\left(\frac{\partial T}{\partial x}\right)^2 + \left(\frac{\partial T}{\partial y}\right)^2 + \left(\frac{\partial T}{\partial z}\right)^2}} = \left(\frac{H_{,i}H_{,i}}{T_{,j}T_{,j}}\right)^{\frac{1}{2}} \tag{1.21}$$

4. Comini method—the apparent heat capacity is approximated by the expression

$$c^* = \frac{1}{n}\left(\frac{\frac{\partial H}{\partial x}}{\frac{\partial T}{\partial x}} + \frac{\frac{\partial H}{\partial y}}{\frac{\partial T}{\partial y}} + \frac{\frac{\partial H}{\partial z}}{\frac{\partial T}{\partial z}}\right) = \frac{1}{n}\frac{H_{,i}}{T_{,i}} \tag{1.22}$$

where n is the number of dimensions.

1.4 Numerical Model of Solidification

Solving the partial differential equations can pass from spatial discretization through time discretization to approximate solution. First, we use the finite element method.

The finite element method facilitates the modeling of many complex problems. Its wide application for founding comes from the fact that it permits an easy adaptation of many existing solutions and techniques of solidification modeling.

Computer calculations need to use discrete models, which means problems must be formulated by introducing time-space mesh. These methods convert given physical equations into matrix equations (algebraic equations). This system of algebraic equations usually contains many thousands of unknowns, that is why the efficiency of a method applied to solve them is crucial.

The discretization of the governing equation in space with the finite element method leads to the ordinary differential equation with time derivative, given as:

$$\mathbf{M}(T)\dot{\mathrm{T}} + \mathbf{K}(T)\mathrm{T} = \mathbf{b}(T) \tag{1.23}$$

where M is the capacity matrix, K is the conductivity matrix, T is the temperature vector and b is the right-hand side vector, whose values are calculated on the boundary conditions basis. The global form of these matrices is obtained by summing the coefficients for all the finite elements. The matrix components are defined for a single finite element as follows:

$$\mathbf{M} = \sum_e \int_{\Omega} c^* \mathbf{N}^T \mathbf{N} \, \mathrm{d}\Omega, \tag{1.24}$$

$$\mathbf{K} = \sum_e \int_{\Omega} \lambda \nabla^T \mathbf{N} \cdot \nabla \mathbf{N} \, \mathrm{d}\Omega, \tag{1.25}$$

$$\mathbf{b} = \sum_e \int_{\Gamma} \mathbf{N}_{\Gamma}^T \mathbf{q}^T \mathbf{n} \, \mathrm{d}\Gamma \,, \tag{1.26}$$

where $\mathbf{N}$ is a shape vector in the area Ω, $\mathbf{N}_{\Gamma}$ is a shape vector on the boundary Γ, $\mathbf{n}$ is an ordinary vector towards the boundary Γ, and $\mathbf{q}$ is a vector of nodal fluxes.

Next, we have applied the time discretization. To do this we used a modified Euler Backward time integration scheme (which belongs to a family of the one-step Θ integration schemes [21]):

$$(\mathrm{M}^n + \Delta t \mathrm{K}^n)\, \mathrm{T}^{n+1} = \mathrm{M}^n \mathrm{T}^n + \Delta t \mathrm{b}^{n+1}, \tag{1.27}$$

where superscript n refers to the number of time step.

1.5 Used Software of Engineering Simulation

Growth of computing power allows engineers to design and run engineering simulations on PC. Researchers can use typical engineering software (some kind of CAD etc.), but some of physical phenomena may not be implemented in such software. Authors decided to write their own solidification computing module, because it made computing each method of heat capacity approximation possible. The module was written in C++ programming language. We used the C++ for the module should be fast, scalable and compatible with chosen numerical utilities.

Both GMSH mesh generator and extended (to include our module) TalyFEM library have been used in numerical experiment [19]. GMSH is a widespread tool which allows finite element mesh of problem geometry (declared or created with graphical interface by user) to be generated. Furthermore, the boundary conditions (surfaces and/or volumes) can be declared with the pre-processor using the graphical user interface.

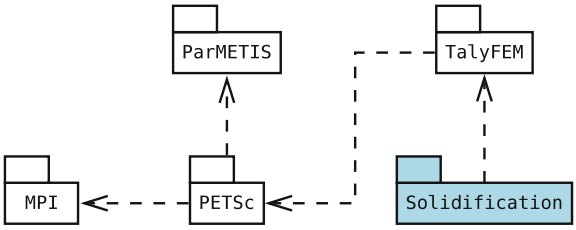

Fig. 1.2 Package diagram in TalyFEM project

TalyFEM is a scalable and extensible framework which uses FEM method to simulate some of physical phenomenon. TalyFEM contains many of PETSc data stuctures, including vectors, matrices or solvers [18]. PETSc library is easy to learn which allows programmers to use mentioned structures in building scalable scientific application. Scalable means using parallel techniques (provided by MPI) but a programmer does not have to write MPI communication routine—all communication about vectors and solvers is written in PETSc and TalyFEM [22]. ParMETIS (for nodes splitting) was also used in our GMSH loading file module (Fig. 1.2). TalyFEM also divides whole problems into sub-meshes (called domains)—each sub-problem can be solved on a separate process (processor or computer node in cluster; Fig. 1.3).

However, the framework does not allow to load meshes generated with GMSH pre-processor. We wrote a GMSH loader module using MPI to communicate the data between processes. Two problems occur:

- loading material properties (which nodes belongs to a cast or an alloy);
- loading neighbourhood nodes (on contact boundary condition) if both nodes are on separate processes;

which have been solved during implementation.

We also implemented solidification module with TalyFEM library. The library requires numerical model of problem solver to be implemented. A programmer does not have to write matrices of the assembly code or communication routines—they are provided with the library. Nevertheless, we considered parameters of boundary conditions (especially contact boundary condition), so we modified standard modules of filling matrices while we were creating the system of linear equations. Framework also solves the created equations (by mentioned communication routines) and writes them into output files (in TecPlot format).

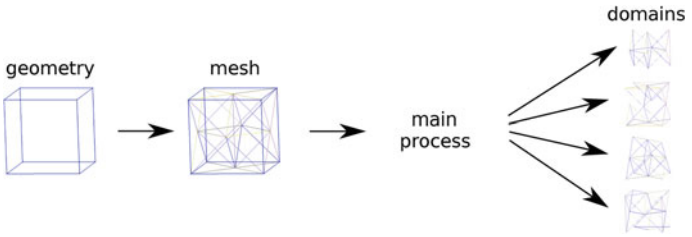

Fig. 1.3 Splitting whole problem to domains in TalyFEM GMSH loader module

1.6 Results of the Numerical Experiment

In the paper we considered a casting solidifying in a metal mold. The finite element mesh comprising 32 814 nodes and 158 417 elements was applied to the area of the casting and mold, as shown in Fig. 1.4. We introduced two boundary conditions: Newton and continuity condition, for which the environment temperature 400 K, the heat transfer coefficient with the environment 10 W/m^2K^{-1} and the heat transfer coefficient through the separation layer 1000 W/m^2K^{-1} are defined. The initial temperature of casting was 960 K, the initial temperature of mold was 600 K, the size of the time step was 0.05 s.

Material properties of the alloy (of which the casting is made of) and the mold are given in Tables 1.1 and 1.2, respectively.

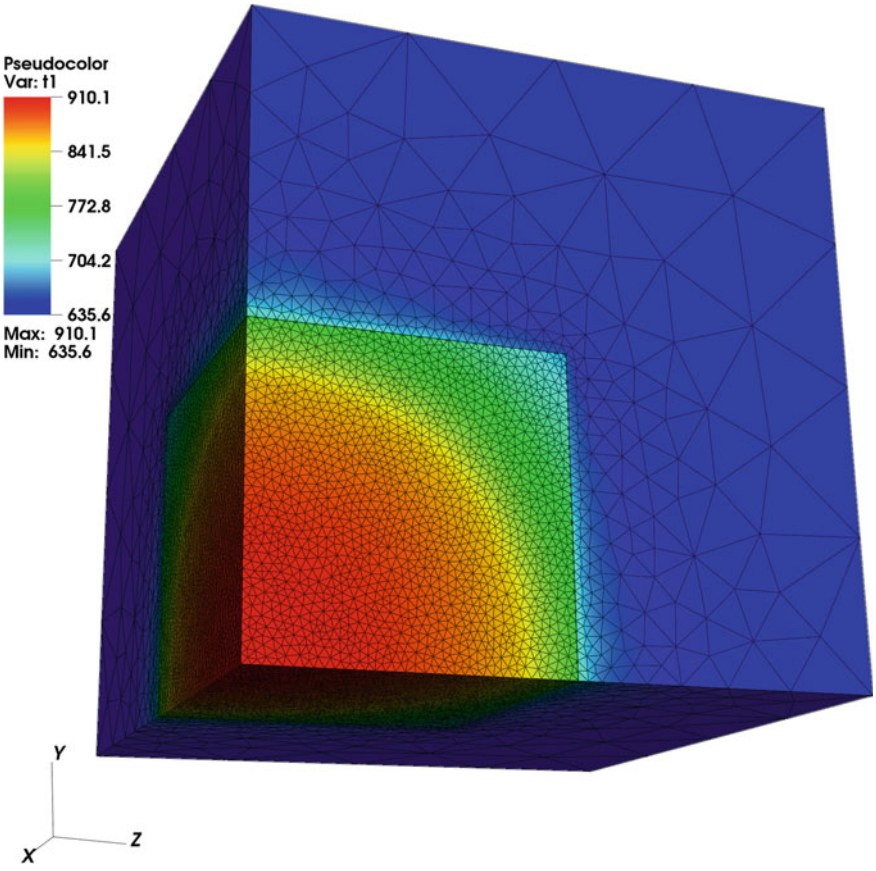

Fig. 1.4 Temperature field after 25 s of cooling. The finite element mesh is also visible

Table 1.1 Material properties of the Al-2 %Cu alloy (subscript s means solid phase and l—liquid phase)

Quantity name	Unit	Value
Density ρ_s	$\frac{\text{kg}}{\text{m}^3}$	2824
Density ρ_l	$\frac{\text{kg}}{\text{m}^3}$	2498
Specific heat c_s	$\frac{\text{J}}{\text{kgK}^{-1}}$	1077
Specific heat c_l	$\frac{\text{J}}{\text{kgK}^{-1}}$	1275
Thermal conductivity λ_s	$\frac{\text{W}}{\text{mK}^{-1}}$	262
Thermal conductivity λ_l	$\frac{\text{W}}{\text{mK}^{-1}}$	104
Solidus temperature T_s	K	853
Liquidus temperature T_l	K	926
Melt temperature of pure component T_M	K	933
Eutectic temperature T_E	K	821
Heat of solidification L	$\frac{\text{J}}{\text{kg}}$	390 000
Partition coefficient of solute k	–	0.125

Table 1.2 Material properties of the mold

Quantity name	Unit	Value
Density	$\frac{\text{kg}}{\text{m}^3}$	7500
The specific heat	$\frac{\text{J}}{\text{kgK}^{-1}}$	620
Thermal conductivity coefficient	$\frac{\text{W}}{\text{mK}^{-1}}$	40

Figure 1.4 shows the distribution of temperatures in the casting after 25 s for the Morgan heat capacity approximation.

The graphs in Figs. 1.5 and 1.7 show the lack of differences in the obtained results. We can see that cooling curves and solid fractions of all methods overlap. It is caused by the fact that although different heat capacity approximations use different formulas, the resulting approximations are very close in values of effective heat capacity, as can be seen in Fig. 1.6.

However, there is a visible difference between the heat capacity approximation formulas in calculation times. The Fig. 1.8 shows the difference in assembly time for different methods. It is easy to notice that the Morgan method requires the least time, while the other formulas are close together in time requirement.

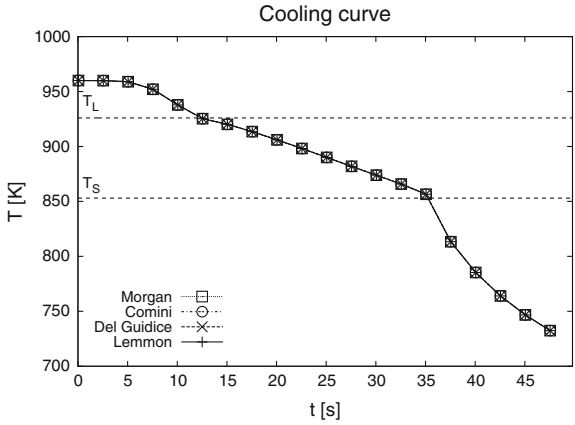

Fig. 1.5 A cooling curve of a point located in origin of coordinate system

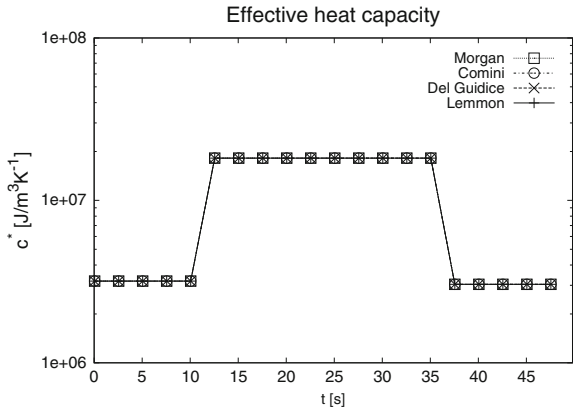

Fig. 1.6 The change of heat capacity approximate during solidification process

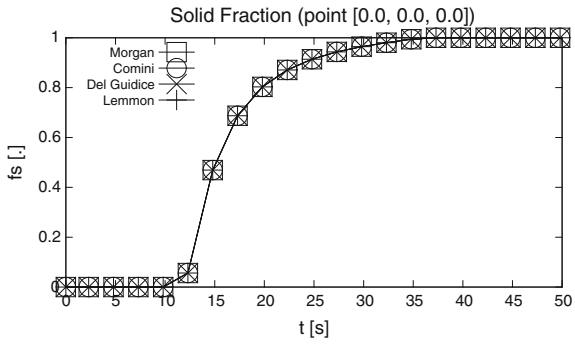

Fig. 1.7 Curve of the solid phase fraction in point (0, 0, 0)

The results from the Fig. 1.8 were obtained for 750 time-steps and mesh from Fig. 1.1. On the 750th time-step (after 37.5 s) the minimum solid fraction was still 0.95 (see Fig. 1.7). This ensures that during the whole calculation time in at least some fraction of finite elements the heat capacity approximation formulas were used.

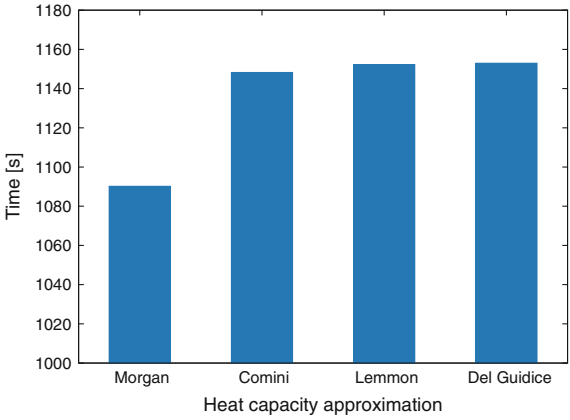

Fig. 1.8 The difference between the heat capacity approximation formulas in time needed for the main matrix assembly

1.7 Conclusions

By analyzing the numerical results obtained from calculations carried out with the help of our own computer program using the finite element method and the apparent heat capacity method we can draw the following remarks and conclusions:

1. The capacity formulation gives an equation very similar to the equation of heat conduction; heat of solidification is hidden in the effective thermal capacity.
2. The use of any of the heat capacity approximation methods does not affect the obtained result, if the solution is stable.
3. When using the Morgan method of heat capacity approximation, one should be careful not to apply too small time step, because then the obtained results might be incorrect.
4. Heat capacity approximation formulas other than Morgan are susceptible to give wrong results if temperature values in nodes of one finite element differ by very small values. The Comini method is especially sensitive to this.
5. Morgan method requires the least time for calculations.

References

1. Stefanescu DM (2002) Science and engineering of casting solidification. Kluwer Academic, New York
2. Wyrzykowski R, Szustak L, Rojek K (2014) Parallelization of 2d mpdata eulag algorithm on hybrid architectures with gpu accelerators. Parallel Comput V40(8):425–447
3. Kim JW, Sandberg RD (2012) Efficient parallel computing with a compact finite difference scheme. Comput Fluids 58:70–87
4. Michalski G, Sczygiol N (2014) Using CUDA architecture for the computer simulation of the casting solidification process. In: Proceedings of the international multiconference of engineers and computer scientists. Lecture notes in engineering and computer science. Hong Kong, 03/2014, pp 933–937

5. Yang N, Li DW, Zhang J, Xi YG (2012) Model predictive controller design and implementation on FPGA with application to motor servo system. Control Eng Pract V20(11):1229–1235
6. Gawronska E, Sczygiol N (2015) Numerically stable computer simulation of solidification: association between eigenvalues of amplification matrix and size of time step. In: Transactions on engineering technologies. Springer, Netherlands, pp 17–30
7. Gawronska E, Sczygiol N (2014) Relationship between eugenvalues and size of time step in computer simulation of thermomechanics phenomena. In: Proceedings of the international multiconference of engineers and computer scientists. Lecture notes in engineering and computer science. Hong Kong, 03/2014, pp. 881–885
8. Ghoneim A, Ojo OA (2011) Numerical modeling and simulation of a diffusion-controlled liquid-solid phase change in polycrystalline solids. Comput Mater Sci V50(3):1102–1113
9. Gawronska E, Sczygiol N (2010) Application of mixed time partitioning methods to raise the efficiency of solidification modeling. In: 12th international symposium on symbolic and numeric algorithms for scientific computing (SYNASC 2010), pp. 99–103
10. Stefanescu F, Neagu G, Mihai A, Stan I, Nicoara M, Raduta A, Opris C (2012) Controlled temperature distribution and heat transfer process in the unidirectional solidification of aluminium alloys. Adv Mater Struct IV 188:314–317
11. Date AW (1994) A novel enthalpy formulation for multidimensional solidification and melting of a pure substance. Sadhana-Acad Proc Eng Sci 19:833–850
12. Duan Q, Tan FL, Leong KC (2002) A numerical study of solidification of n-hexadecane based on the enthalpy formulation. J Mater Process Technol V120(1/3):249–258
13. Ganguly S, Chakraborty S (2006) A generalized formulation of latent heat functions in enthalpy—based mathematical models for multicomponent alloy solidification systems. Metall Mater Trans B-Process Metall Mater Process Sci V37(1):143–145
14. Prabha AJ, Raju S, Jeyaganesh B, Rai AK et al (2011) Thermodynamics of $\alpha'' \rightarrow \alpha$ phase transformation and heat capacity measurements in Ti-15at%Nb alloy. Phys B V406(22):4200–4209
15. Famouri M, Jannatabadi M, Ardakani HTF (2013) Simultaneous estimations of temperature-dependent thermal conductivity and heat capacity using a time efficient novel strategy based on megann. Appl Soft Comput V13(1):201–210
16. Dyja R, Sczygiol N, Domanski Z, Ao S, Chan A, Katagiri H, Xu L (2014) The effect of cavity formation on the casting heat dissipation rate. In: IAENG transactions on engineering sciences, pp 341–347
17. Dyja R, Gawronska E, Grosser A, Jeruszka P, Sczygiol N (2015) Comparison of different heat capacity approximation in solidification modeling. In: Proceedings of the world congress on engineering and computer science 2015. Lecture notes in engineering and computer science. San Francisco, USA, pp 875-879. Accessed 21–23 Oct 2015
18. Balay S et al (2014) PETSc Users Manual, Argonne National Laboratory
19. Kodali HK, Ganapathysubramanian B (2012) A computational framework to investigate charge transport in heterogeneous organic photovoltaic devices. Comput Methods Appl Mech Eng N247:113–129
20. Sczygiol N, Szwarc G (2001) Application of enthalpy formulation for numerical simulation of castings solidification. Comput Assist Mech Eng Sci N8:99–120
21. Wood LW (1990) Practical time-stepping schemes. Clarendon Press, Oxford
22. MPI: A Message Passing Interface. http://www.mpi-forum.org, mpi30-report.pdf, 15.03.2016 r

Chapter 2
Analysis of Systemic Risk: A Dynamic Vine Copula-Based ARMA-EGARCH Model

Kuan-Heng Chen and Khaldoun Khashanah

2.1 Introduction

The definition of systemic risk from the report to G20 Finance Ministers and Governors agreed upon among the International Monetary Fund (IMF), Bank for International Settlements (BIS), and Financial Stability Board (FSB) [1] is that "(i) caused by an impairment of all or parts of the financial system and (ii) has the potential to have serious negative consequences for the real economy". Furthermore, "G-20 members consider an institution, market or instrument as systemic if its failure or malfunction causes widespread distress, either as a direct impact or as a trigger for broader contagion." A common factor from the various definitions of systemic risk is that a trigger event causes a chain of adverse economic consequences, referred to as a "domino effect". Given the definition of systemic risk quoted above, measuring systemic risk is done by estimating the probability of failure of an institution that is the cause of a distress for the financial system. Therefore, we only consider the Value-at-Risk (VaR), the potential loss in value of an asset or portfolio for a given time period and probability, as the risk measurement. In addition, the VaR ratio of a sector to the system (S&P 500 Index), which interprets that the sector risk provides to the entire system, is present.

Girardi and Ergün [2] modified the CoVaR methodology that is proposed by Adrian and Brunnermeier [3] which used the dynamic conditional correlation GARCH, while Hakwa et al. [4] modified the methodology based on copula

K.-H. Chen (✉) · K. Khashanah
Department of Financial Engineering, Stevens Institute of Technology,
Hoboken, NJ 07030, USA
e-mail: kchen3@stevens.edu

K. Khashanah
e-mail: kkhashan@stevens.edu

S.-I. Ao et al. (eds.), *Transactions on Engineering Technologies*,
DOI 10.1007/978-981-10-2717-8_2

modeling. We present dynamic vine Copula-based ARMA-EGARCH (1, 1) VaR measure into a high dimensional analysis in systemic risk.

Sklar [5] introduced the copula to describe the dependence structure between variables. Patton [6] defined the conditional version of Sklar's theorem, which extends the copula applications to the time series analysis. Otani and Imai [7] presented a basket credit default swaps (CDSs) pricing model with nested Archimedean copulas. However, multivariate Archimedean copulas are limited in that there are only one or two parameters to capture the dependence structure. Joe [8] introduced a construction of multivariate distribution based on pair-copula construction (PCC), while Aas et al. were the first to recognize that the pair-copula construction (PCC) principle can be used with arbitrary pair-copulas, referred to as the graphical structure of R-vines [9]. Furthermore, Dissmann et al. [10] developed an automated algorithm of jointly searching for an appropriate R-vines tree structures, the pair-copula families and their parameters. Accordingly, a high dimensional joint distribution can be decomposed to bivariate and conditional bivariate copulas arranged together according to the graphical structure of a regular vine. Besides, Rockinger and Jondeau [11] was the first to introduce the copula-based GARCH modeling. Afterwards, Lee and Long [12] concluded that copula-based GARCH models outperform the dynamic conditional correlation model, the varying correlation model and the BEKK model. In addition, Fang et al. [13] investigated that using Akaike Information Criterion (AIC) as a tool for choosing copula from a couple of candidates is more efficient and accurate than the multiplier goodness-of-fit test method.

During 2008, the subprime mortgage crisis was a systemic collapse triggered by the financial industry. The purpose of this paper is to present an application of the estimation of systemic risk in terms of the VaR/ES ratio by using the dynamic vine copula-based ARMA-EGARCH (1, 1) model. To compute systemic risk for our system, we use S&P 500 sector indices and S&P 500 index to be our components and system, respectively. Since the parameters change over time, we calibrate the parameters every ten steps to capture a change of the structure. This scalable prototype of US financial system with limited dimensionality can be easily tailored to any underlying sector, country or financial market.

This paper has four sections. The first section briefly introduces existing research regarding systemic risk. The second section describes the definition of the VaR/ES ratio, and outlines the methodology of vine Copula-based EGARCH (1, 1) modeling. The third section describes the data and explains the results of VaR/ES ratio. The fourth section concludes our findings.

2.2 Methodology

2.2.1 Risk Methodology

The definition of Value-at-Risk (VaR) is that the maximum loss at most is (1-α) probability given by a period [14]. People usually determine α as 95 %, 99 %, or 99.9 % to be their confidence level. In this study, we use the Copula-based ARMA-EGARCH (1, 1) methodology to obtain the dynamic VaR from each sector. We denote $VaR^{i\to j}_{t,1-\alpha}$ ratio, the sector i's risk contribution to the system j (S&P 500 index) at the confidence level α, by

$$VaR^{i\to j}_{t,1-\alpha}\,Ratio = \frac{VaR^{i}_{t,1-\alpha}}{VaR^{j}_{t,1-\alpha}}$$

The higher $VaR^{i\to j}_{t,1-\alpha}$ ratio indicates that the sector is the risk provider to the system. In addition, the methodology can be easily extended from the VaR ratio to the expected shortfall (ES) ratio.

2.2.2 Univariate ARMA-EGARCH Model

Engle is the first researcher to introduce the ARCH model, which deals with volatility clustering, usually referred to as conditional heteroskedasticity. Bollerslev [15] extended the ARCH model to the generalized ARCH (GARCH) model. Chen and Khashanah [16] implemented ARMA (p, q)-GARCH (1, 1) with the Student's t distributed innovations for the marginal to account for the time-varying volatility, whereas the Student's t distributed innovations cannot explain the skewness. In addition, to overcome the leverage effect in financial time series, we use the exponential GARCH (EGARCH) model in handling asymmetric effects between positive and negative asset returns proposed by Nelson [17]. According to the augmented Dickey–Fuller (ADF) test, all the series are stationary. Therefore, ARMA (p, q)-EGARCH (1, 1) with the skewed Student's t distributed innovation can then be written as [18]

$$\begin{aligned} r_t &= \mu_t + \sum_{i=1}^{p} \vartheta_i r_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t, \\ \varepsilon_t &= \sigma_t z_t, \\ \ln(\sigma_t^2) &= \kappa_t + \alpha_t z_{t-1} + \xi_t(|z_{t-1}| - \mathrm{E}\{|z_{t-1}|\}) + \beta_t \ln(\sigma_{t-1}^2) \end{aligned}$$

where r_t is the log return, μ_t is the drift term, ε_t is the error term, ξ_t capture the size effect, and the standardized innovation term z_t is the skewed Student's t distribution. The skewed student's t density function can be expressed as [19]

$$p(z|\eta,f)=\frac{2}{\eta+\eta^{-1}}\left\{f\left(\frac{z}{\eta}\right)I_{[0,\infty)}(z)+f(\eta z)I_{(-\infty,0]}(z)\right\}$$

where f is a univariate pdf that is symmetric around 0, I_S is the indicator function on S, η is the asymmetric parameter, and $\eta=1$ for the symmetric Student's t distribution. In addition, the correlated random variables can be flexible and easily estimated under an overwhelming feature of Copula-based ARMA-EGARCH model.

2.2.3 Sklar's Theory

Sklar's Theorem [5] states that given random variables $X_1, X_2, \ldots, X_n$ with continuous distribution functions $F_1, F_2, \ldots, F_n$ and joint distribution function H, and there exists a unique copula C such that for all $x=(x_1, x_2, \ldots, x_n)\in\Re^n$

$$H(x)=C(F_1(x_1), F_2(x_2), \ldots, F_n(x_n))$$

If the joint distribution function is n-times differentiable, then taking the nth cross-partial derivative of the equation:

$$\begin{aligned}f(x_1, x_2, \ldots, x_n)&=\frac{\partial^n}{\partial x_1\ldots\partial x_n}H(x)\\&=\frac{\partial^n}{\partial u_1\ldots\partial u_n}C(F_1(x_1), \ldots, F_n(x_n))\cdot\prod_{i=1}^{n}f_i(x_i)\\&=c(F_1(x_1), \ldots, F_n(x_n))\cdot\prod_{i=1}^{n}f_i(x_i)\end{aligned}$$

where u_i is the probability integral transform of x_i.

For the purpose of estimating the VaR or ES based on time series data, Patton [6] defined the conditional version of Sklar's theorem. Let $F_{1,t}$ and $F_{2,t}$ be the continuous conditional distributions of $x_1|\mathfrak{F}_{t-1}$ and $x_2|\mathfrak{F}_{t-1}$, given the conditioning set $\mathfrak{F}_{t-1}$, and let H_t be the joint conditional bivariate distribution of $(X_1, X_2|\mathfrak{F}_{t-1})$. Then, there exists a unique conditional copula C_t such that

$$H_t(x_1, x_2|\mathfrak{F}_{t-1})=C_t(F_{1,t}(x_1|\mathfrak{F}_{t-1}), F_{2,t}(x_2|\mathfrak{F}_{t-1})|\mathfrak{F}_{t-1})$$

2.2.4 Parametric Copulas

Joe [9] and Nelsen [20] gave comprehensive copula definitions for each family.

(1) The bivariate Gaussian copula is defined as:

$$C(u_1, u_2; \rho) = \Phi_\rho(\Phi^{-1}(u_1), \Phi^{-1}(u_2))$$

where Φ_ρ is the bivariate joint normal distribution with linear correlation coefficient ρ and Φ is the standard normal marginal distribution.

(2) The bivariate student's t copula is defined by the following:

$$C(u_1, u_2; \rho, \nu) = t_{\rho,\nu}(t_\nu^{-1}(u_1), t_\nu^{-1}(u_2))$$

where ρ is the linear correlation coefficient and ν is the degree of freedom.

(3) The Clayton generator is given by $\varphi(u) = u^{-\theta} - 1$ with $\theta \in (0, \infty)$, its copula is defined by

$$C(u_1, u_2; \theta) = (u_1^{-\theta} + u_2^{-\theta} - 1)^{-1/\theta}$$

(4) The Gumbel generator is given by $\varphi(u) = (-\ln u)^\theta$ with $\theta \in [1, \infty)$, and the bivariate Gumbel copula is given by

$$C(u_1, u_2; \theta) = \exp(-\left\{(-\ln u_1)^\theta + (-\ln u_2)^\theta\right\}^{1/\theta})$$

(5) The Frank generator is given by $\varphi(u) = \ln[(e^{-\theta u} - 1)/(e^{-\theta} - 1)]$ with $\theta \in (-\infty, 0) \cup (0, \infty)$, and the bivariate Frank copula is defined by

$$C(u_1, u_2; \theta) = -\frac{1}{\theta}\ln(1 + \frac{\left(e^{-\theta u_1} - 1\right)\left(e^{-\theta u_2} - 1\right)}{e^{-\theta} - 1})$$

(6) The Joe generator is $\varphi(u) = u^{-\theta} - 1$, and the Joe copula is given by

$$C(u_1, u_2; \theta) = 1 - (\overline{u_1}^\theta + \overline{u_2}^\theta - \overline{u_1}^\theta \overline{u_2}^\theta)^{1/\theta}, \text{with } \theta \in [1, \infty)$$

(7) The BB1 (Clayton-Gumbel) copula with $\theta \in (0, \infty) \cap \delta \in [1, \infty)$ is

$$C(u_1, u_2; \theta, \delta) = \left(1 + \left[\left(u_1^{-\theta} - 1\right)^{\delta} + \left(u_2^{-\theta} - 1\right)^{\delta}\right]^{1/\delta}\right)^{-1/\theta}$$

(8) The BB6 (Joe-Gumbel) copula with $\theta \in [1, \infty) \cap \delta \in [0, \infty)$ is

$$C(u_1, u_2; \theta, \delta) = 1 - \left(1 - \exp\left\{-\left[\left(-\ln\left(1 - \overline{u}_1^{\theta}\right)\right)^{\delta} + \left(-\ln\left(1 - \overline{u}_2^{\theta}\right)\right)^{\delta}\right]^{1/\delta}\right\}\right)^{1/\theta}$$

(9) The BB7 (Joe-Clayton) copula with $\theta \in [1, \infty) \cap \delta \in [0, \infty)$ is

$$C(u_1, u_2; \theta, \delta) = 1 - \left(1 - \left[\left(1 - \overline{u}_1^{\theta}\right)^{-\delta} + \left(1 - \overline{u}_2^{\theta}\right)^{-\delta} - 1\right]^{-1/\delta}\right)^{1/\theta}$$

(10) The BB8 (Frank-Joe) copula with $\theta \in [1, \infty) \cap \delta \in (0, 1]$ is

$$C(u_1, u_1; \theta, \delta) = \frac{1}{\delta}\left(1 - \left[1 - \frac{1}{1 - (1 - \delta)^{\theta}}\left(1 - (1 - \delta u_1)^{\theta}\right)\left(1 - (1 - \delta u_2)^{\theta}\right)\right]^{1/\theta}\right)$$

2.2.5 Vine Copulas

Even though it is simple to generate multivariate Archimedean copulas, they are limited in that there are only one or two parameters to capture the dependence structure. Vine copula method allows a joint distribution to be built from bivariate and conditional bivariate copulas arranged together according to the graphical structure of a regular vine, which is a more flexible measure to capture the dependence structure among assets. It is well known that any multivariate density function can be decomposed as

$$f(x_1, \ldots, x_n) = f(x_n | x_1, \ldots, x_{n-1}) \ldots f(x_3 | x_1, x_2) f(x_2 | x_1) f(x_1)$$

Moreover, the conditional densities can be written as copula functions. For instance, the first and second conditional density can be decomposed as

$$
\begin{aligned}
f(x_2|x_1) &= c_{1,2}(F_1(x_1), F_2(x_2)) \cdot f_2(x_2), \\
f(x_3|x_1, x_2) &= c_{2,3|1}(F_{2|1}(x_2|x_1), F_{3|1}(x_3|x_1)) \cdot f_3(x_3|x_1) \\
&= c_{2,3|1}(F_{2|1}(x_2|x_1), F_{3|1}(x_3|x_1)) \cdot c_{1,3}(F_1(x_1), F_3(x_3)) \cdot f_3(x_3)
\end{aligned}
$$

After rearranging the terms, the joint density can be written as

$$
\begin{aligned}
f(x_1, x_2, x_3) = c_{2,3|1}(F_{2|1}(x_2|x_1), F_{3|1}(x_3|x_1)) \cdot c_{1,2}(F_1(x_1), F_2(x_2)) \\
\cdot c_{1,3}(F_1(x_1), F_3(x_3)) \cdot f_1(x_1) \cdot f_2(x_2) \cdot f_3(x_3)
\end{aligned}
$$

The summary of vine copulas is given by Kurowicka and Joe [21], and the general n-dimensional canonical vine copula, in which one variable plays a pivotal role, can be written as

$$
f(x_1, \ldots, x_n) = \prod_{k=1}^{n} f(x_k) \times \prod_{j=1}^{n-1} \prod_{i=1}^{n-j} c_{j,j+i|1,\ldots,j-1}
$$

Similarly, D-vines are constructed by choosing a specific order for the variables, and the general n-dimensional D-vine copula can be written as

$$
f(x_1, \ldots, x_n) = \prod_{k=1}^{n} f(x_k) \times \prod_{j=1}^{n-1} \prod_{i=1}^{n-j} c_{i,i+j|i+1,\ldots,i+j-1}
$$

Dissmann et al. [10] proposed that the automated algorithm involves searching for an appropriate R-vine tree structure, the pair-copula families, and the parameter values of the chosen pair-copula families based on AIC, which is summarized in Table 2.1.

Table 2.1 Sequential method to select an R-Vine model

Algorithm. Sequential method to select an R-Vine model
1. Calculate the empirical Kendall's tau for all possible variable pairs.
2. Select the tree that maximizes the sum of absolute values of Kendall's taus.
3. Select a copula for each pair and fit the corresponding parameters based on AIC.
4. Transform the observations using the copula and parameters from Step 3. To obtain the transformed values.
5. Use transformed observations to calculate empirical Kendall's taus for all possible pairs.
6. Proceed with Step 2. Repeat until the R-Vine is fully specified.