M. Kunick, H.-J. Kretzschmar
Zittau/Görlitz University of Applied Sciences, Department of Technical Thermodynamics, Zittau, Germany

U. Gampe
Technical University of Dresden, Institute for Power Engineering, Chair of Thermal Power Machinery and Plants, Dresden, Germany

Fast Calculation of Thermodynamic Properties of Water and Steam in Process Modelling using Spline Interpolation

Agenda

- Introduction: Motivation and Aims
- Spline Interpolation of Thermodynamic Properties
- Example: Spline Function $T_2^{\mathrm{SPL}}(p,h)$ and Inverse Spline Function $h_2^{\mathrm{INV}}(p,T)$
- Computing Time comparisons with IAPWS-IF97 and TTSE
- Summary and Outlook

Introduction: Motivation and Aims

Algorithms for calculating thermodynamic properties used in process modelling have to fulfill these requirements:

- → Thermodynamic properties must be represented continuously with high accuracy
- → Numerical consistency between forward and backward functions
- → High computing speed

Current state of fast property algorithms:

- → IAPWS-IF97 contains fast fundamental equations and backward equations.
- → Table look-up methods: IAPWS adopted Tabular Taylor Series Expansion Method (TTSE) as a guideline in 2003. TTSE is not able to represent thermodynamic property functions continuously.

Further requirements of Computational Fluid Dynamics (CFD) and calculations of non-stationary processes on property calculations:

- → Extremely high numerical consistency between forward and backward functions
- → Extremely high computing speed
- → Flexible algorithm, suitable for property functions such as $p(v,h)$ and $p(v,u)$

⇓

Can table look-up methods meet these requirements?

Spline Interpolation of Thermodynamic Properties

Characteristics of spline functions:

- → Consists of piecewise defined functions (spline polynomials)
- → Continuous and smooth representation of one- or multi-dimensional functions
- → Several kinds of splines and ways to determine them are possible

Choice of spline function:

- → Demand on high computing speed leads to simple spline polynomials
- → Bi-quadratic spline polynomial can be solved in terms of its variables analytically
- → On piecewise equidistant grids simple search algorithm can be applied
- → Special data handling can be applied which enables high computing speed

Bi-quadratic polynomial:

$$z_{ij}^{\mathrm{SPL}}\left(x_1, x_2\right) = \sum_{k=1}^{3}\sum_{l=1}^{3} a_{ijkl}\left(x_1 - x_{1i}\right)^{k-1}\left(x_2 - x_{2j}\right)^{l-1}$$

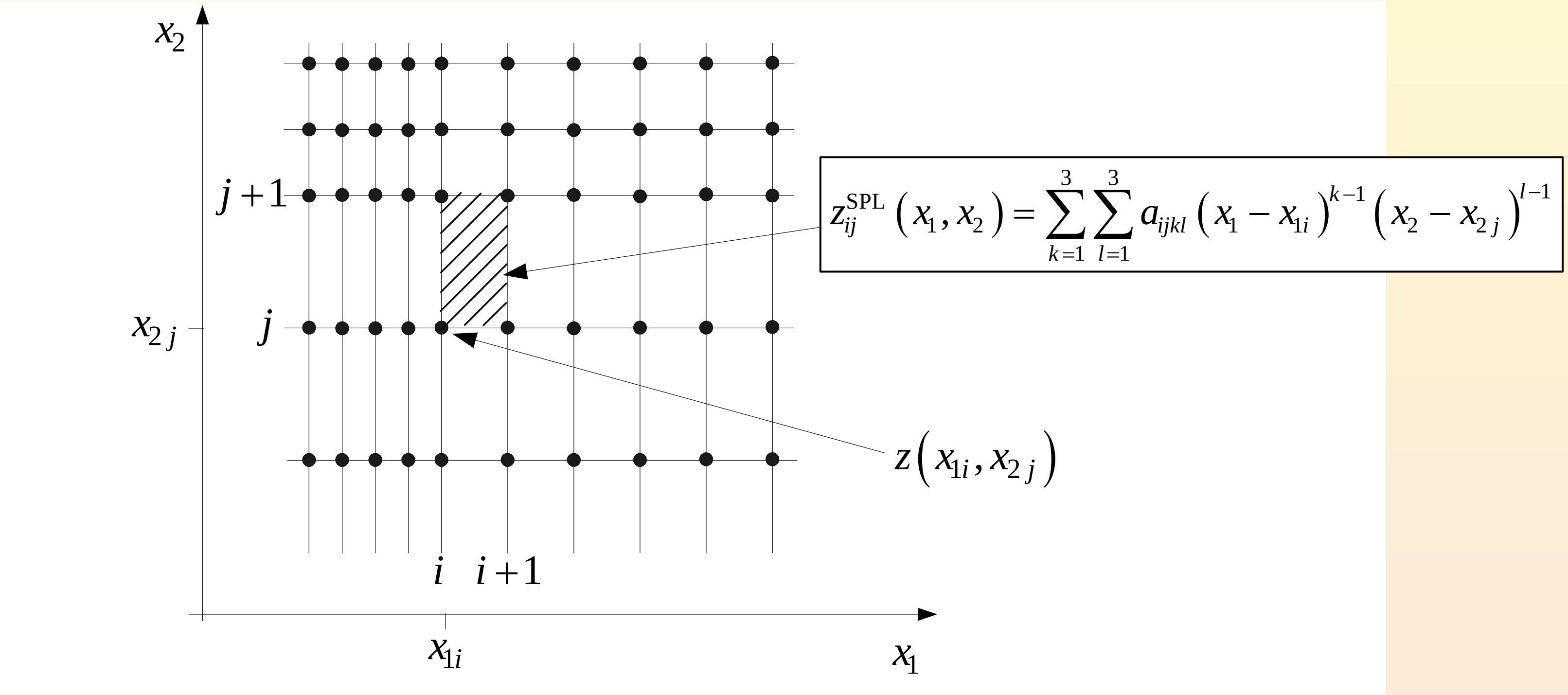
$$\begin{aligned} z_{ij}^{\mathrm{SPL}}\left(x_1, x_2\right) = {} & a_{ij11} + a_{ij21}\Delta x_{1i} + a_{31}\Delta x_{1i}^2 \\ & + a_{ij12}\Delta x_{2j} + a_{ij22}\Delta x_{1i}\Delta x_{2j} + a_{32}\Delta x_{1i}^2\Delta x_{2j} \\ & + a_{ij13}\Delta x_{2j}^2 + a_{ij23}\Delta x_{1i}\Delta x_{2j}^2 + a_{33}\Delta x_{1i}^2\Delta x_{2j}^2 \end{aligned}$$

with $\Delta x_{1i} = \left(x_1 - x_{1i}\right)$ and $\Delta x_{2j} = \left(x_2 - x_{2j}\right)$

Spline Interpolation of Thermodynamic Properties

Creating a spline function $z^{\mathrm{SPL}}(x_1, x_2)$ from a given e.o.s. $z(x_1, x_2)$:

- → Calculation of a piecewise equidistant grid of nodes from a given equation of state $z(x_1, x_2)$
- → Determination of all spline coefficients for each function $z_{ij}^{\mathrm{SPL}}(x_1, x_2)$
- → Several ways are possible (depending on interpolation conditions)



Spline Interpolation of Thermodynamic Properties

Inverse spline functions:

- Bi-quadratic polynomials can be solved analytically in terms of its variables
- Inverse spline function:

$$x_{1,ij}^{\mathrm{INV}}\left(z, x_2\right) = \frac{\left(-B \pm \sqrt{B^2 - 4AC}\right)}{2A} + x_{1i}$$

with $A = a_{ij31} + \Delta x_{2j}\left(a_{ij32} + a_{ij33}\Delta x_{2j}\right)$ $B = a_{ij21} + \Delta x_{2j}\left(a_{ij22} + a_{ij23}\Delta x_{2j}\right)$ $C = a_{ij11} + \Delta x_{2j}\left(a_{ij12} + a_{ij13}\Delta x_{2j}\right) - z$

and $\Delta x_{2j} = \left(x_2 - x_{2j}\right)$

- The inverse spline function is completely numerical consistent to the spline function $z_{ij}^{\mathrm{SPL}}\left(x_1, x_2\right)$

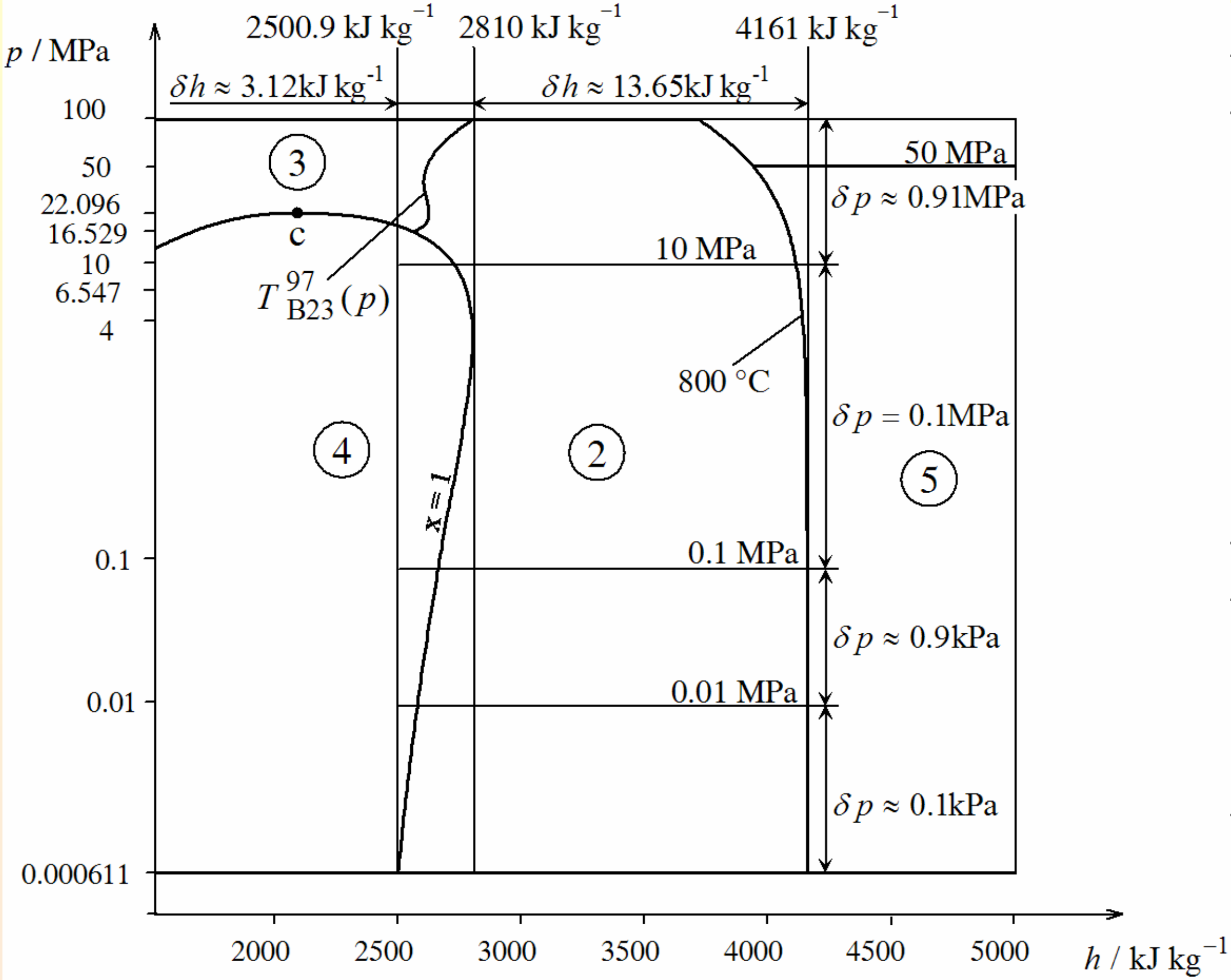
Computing speed:

- Mathematical operations such as square root and division are more time consuming than multiplications or additions
- A more time consuming search algorithm is necessary to determine the sub-rectangle from given (z, x_2)
- Therefore the computing speed of an inverse spline function is lower than the speed of the corresponding spline function

Example: Spline Function $T_2^{\mathrm{SPL}}(p,h)$ and Inverse Spline Function $h_2^{\mathrm{INV}}(p,T)$

The *p-h* grid:

→ In order to reach high accuracy (5 significant figures) the grid was created as shown here:



Range	*p*-grid lines
$0.000611\ \mathrm{MPa} \le p \le 0.01\ \mathrm{MPa}$	100
$0.01\ \mathrm{MPa} \le p \le 0.1\ \mathrm{MPa}$	100
$0.1\ \mathrm{MPa} \le p \le 10\ \mathrm{MPa}$	100
$10\ \mathrm{MPa} \le p \le 100\ \mathrm{MPa}$	100

Range	*h*-grid lines
$2500.9\ \mathrm{kJ\,kg^{-1}} \le h \le 2810\ \mathrm{kJ\,kg^{-1}}$	50
$2810\ \mathrm{kJ\,kg^{-1}} \le h \le 4161\ \mathrm{kJ\,kg^{-1}}$	100

Computing Time comparisons to IAPWS-IF97 and to TTSE

→ Using the IAPWS-software NIFBENCH computing time comparisons have been carried out

Spline function $T_2^{\mathrm{SPL}}(p,h)$:

→ Computing times was compared to $T_2^{\mathrm{97BW}}(p,h)$ and $T^{\mathrm{TTSE}}(p,h)$

→ Computing times in μs:

$T_2^{\mathrm{SPL}}(p,h)$	$T_2^{\mathrm{97BW}}(p,h)$	$T^{\mathrm{TTSE}}(p,h)$
0.056	0.114	0.178

→ The spline function $T_2^{\mathrm{SPL}}(p,h)$ is 2 times faster than the backward equation $T_2^{\mathrm{97BW}}(p,h)$

Inverse spline function $h_2^{\mathrm{INV}}(p,T)$:

→ Computing times was compared to $h_2^{\mathrm{IF97}}(p,T)$ and $h^{\mathrm{TTSE}}(p,T)$

→ Computing times in μs:

$h_2^{\mathrm{INV}}(p,T)$	$h_2^{\mathrm{IF97}}(p,T)$	$h^{\mathrm{TTSE}}(p,T)$
0.202	0.242	0.237

→ The inverse spline function $h_2^{\mathrm{INV}}(p,T)$ is 1.2 times faster than $h_2^{\mathrm{IF97}}(p,T)$

→ The inverse spline function $h_2^{\mathrm{INV}}(p,T)$ is completely numerically consistent to $T_2^{\mathrm{SPL}}(p,h)$

Summary

- Spline functions are able to represent thermodynamic properties continuously
- A spline function $T_2^{\mathrm{SPL}}\left(p,h\right)$ has been created in a first study
  - High accuracy (5 significant figures) could be achieved
  - $T_2^{\mathrm{SPL}}\left(p,h\right)$ is 2 times faster than the corresponding backward equation $T_2^{\mathrm{97BW}}\left(p,h\right)$
- By solving $T_2^{\mathrm{SPL}}\left(p,h\right)$ in terms of h, the inverse spline function $h_2^{\mathrm{INV}}\left(p,T\right)$ could be obtained
  - The inverse spline function $h_2^{\mathrm{INV}}\left(p,T\right)$ is completely numerically consistent to $T_2^{\mathrm{SPL}}\left(p,h\right)$

Outlook

- The algorithm will be modified for non-rectangular grids
- An algorithm for grid optimization is being developed
- Other spline functions will be investigated
- The algorithm will be extended for mixtures
- Software for automatic generation of spline functions will be developed
  - From a given equation of state spline functions will be generated
  - For the desired range of validity and desired accuracy a spline function will be optimized
  - Source code will be generated automatically

Thank you for paying attention.