M. Kunick, H.-J. Kretzschmar
Zittau/Goerlitz 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 in Process Modeling Using Spline-Interpolation

Agenda

- Calculation of Thermodynamic Properties: Requirements and Algorithms
- Spline-Interpolation of Thermodynamic Properties
- Example 1: A Set of Spline-Functions for Gaseous CO_2
- Example 2: Comparison of a Spline-Function $T_2^{\mathrm{SPL}}(p,h)$ with the Corresponding Backward-Equation $T_2^{\mathrm{IF97}}(p,h)$ for Region 2 of IAPWS-IF97 for Steam
- Software-Tool for Generating Spline-Based Property Libraries
- Summary and Outlook

Designing and Optimizing Advanced Power Cycles – Calculation of Thermodynamic Properties: Requirements and Algorithms

Requirements	Precise fundamental equations	Fast fundamental equations and backward equations	Table look-up methods
Accuracy	high	sufficient for industrial use	depends on table size and interpolation algorithm
Computing speed	low, because: • contain numerous transcendental terms • backward functions calculated iteratively	high, because: • contain integer exponents only (calculable using Horner's scheme) • backward functions calculated without iterations	very high, because: • simple interpolation equations are in use • backward functions calculated without iterations
Numerical consistency	depending on number of iterations	limited	complete numerical consistency is possible
Availability	available for most working fluids	for only a few working fluids backward equations are available	tables must be prepared in advance

Developments in computer technology

- Improvement of clock rate of CPU's stagnates
- Extremely large main memory available now

Advantages of Spline-Interpolation

- Continuous representation of one- or multi-dimensional functions and their derivatives
- High accuracy with small tables (grids) in comparison to linear interpolation
- Spline polynomials up to third degree can be solved analytically in terms of their variables

Spline-based table look-up methods are preferable.

Spline Interpolation of Thermodynamic Properties

Choice of spline polynomial and calculation method:

- → Demand for high computing speed leads to simple spline polynomials.
- → Spline polynomials up to third degree can be solved analytically in terms of their variables. This enables complete numerical consistency between forward and backward functions.
- → Example: bi-quadratic polynomial (to be defined in each cell (*i*,*j*))

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

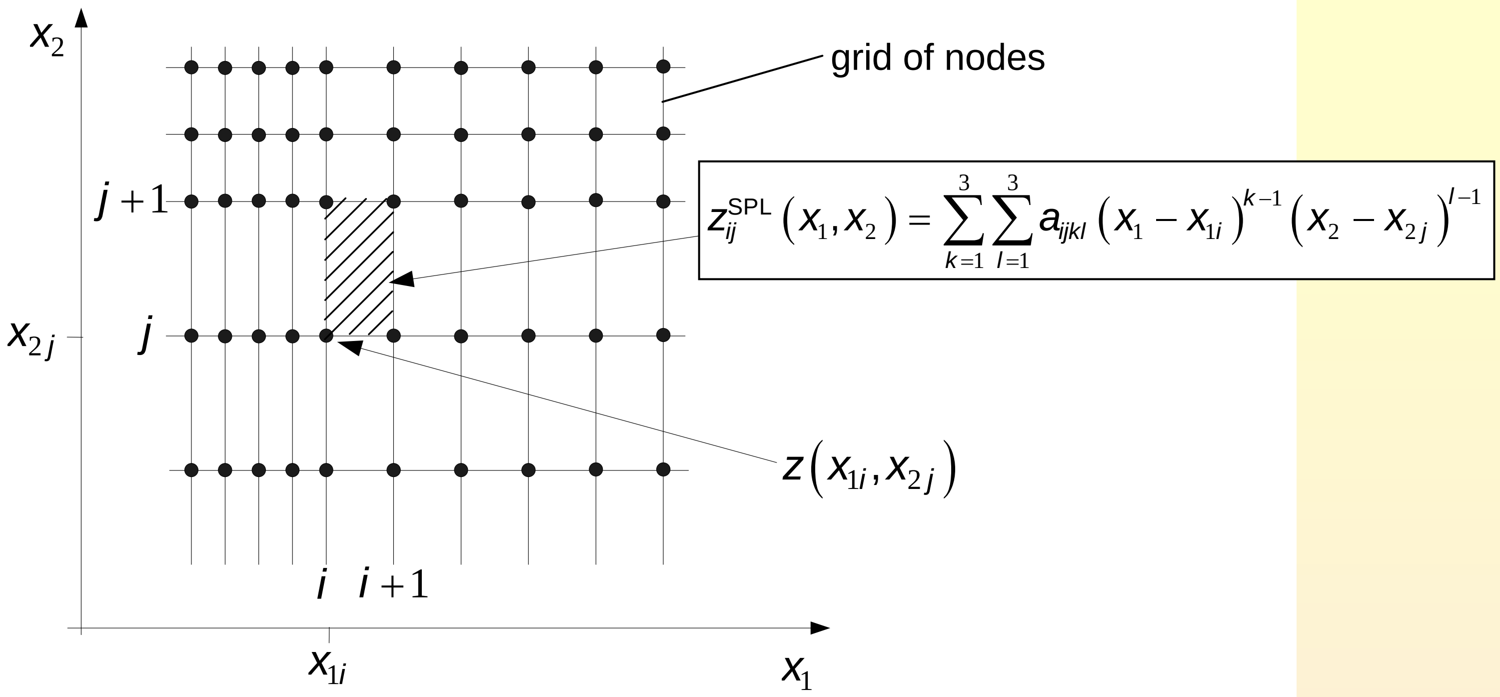
complete notation:

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

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

- → Several calculation methods for determining the coefficients lead to different kinds of splines with differing characteristics.

Creating a spline function $z^{SPL}(x_1,x_2)$ from a given EOS $z^{EOS}(x_1,x_2)$:



→ Generation of a piecewise equidistant grid of nodes:

- density of nodes can be adjusted to the curvature of the property surface locally for desired accuracy
- simple search algorithms can be applied
- enables fast data handling which is important for high computing speed

→ Calculation of required function values and derivatives at all nodes from a given equation of state $z^{EOS}(x_1,x_2)$.

→ Determination of all spline coefficients a_{ijkl} for each polynomial $z_{ij}^{SPL}(x_1,x_2)$.

→ Store all coefficients in a look-up table.

Calculation of inverse spline functions:

→ Bi-quadratic polynomials can be solved analytically in terms of their variables

→ Inverse spline function $x_{1,ij}^{\mathrm{INV}}\left(z,x_2\right)$ of cell (i,j):

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

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 numerically consistent with the spline function $z_{ij}^{\mathrm{SPL}}\left(x_1,x_2\right)$.

Set of spline functions defined in a *p*-*h* grid:

- In process modeling properties are often calculated from (p,h).
- Spline Functions of p and h: $T^{SPL}(p,h)$ $s^{SPL}(p,h)$ $v^{SPL}(p,h)$ …
- Calculation from other pairs of variables using these spline functions:
- Inverse Spline Functions:

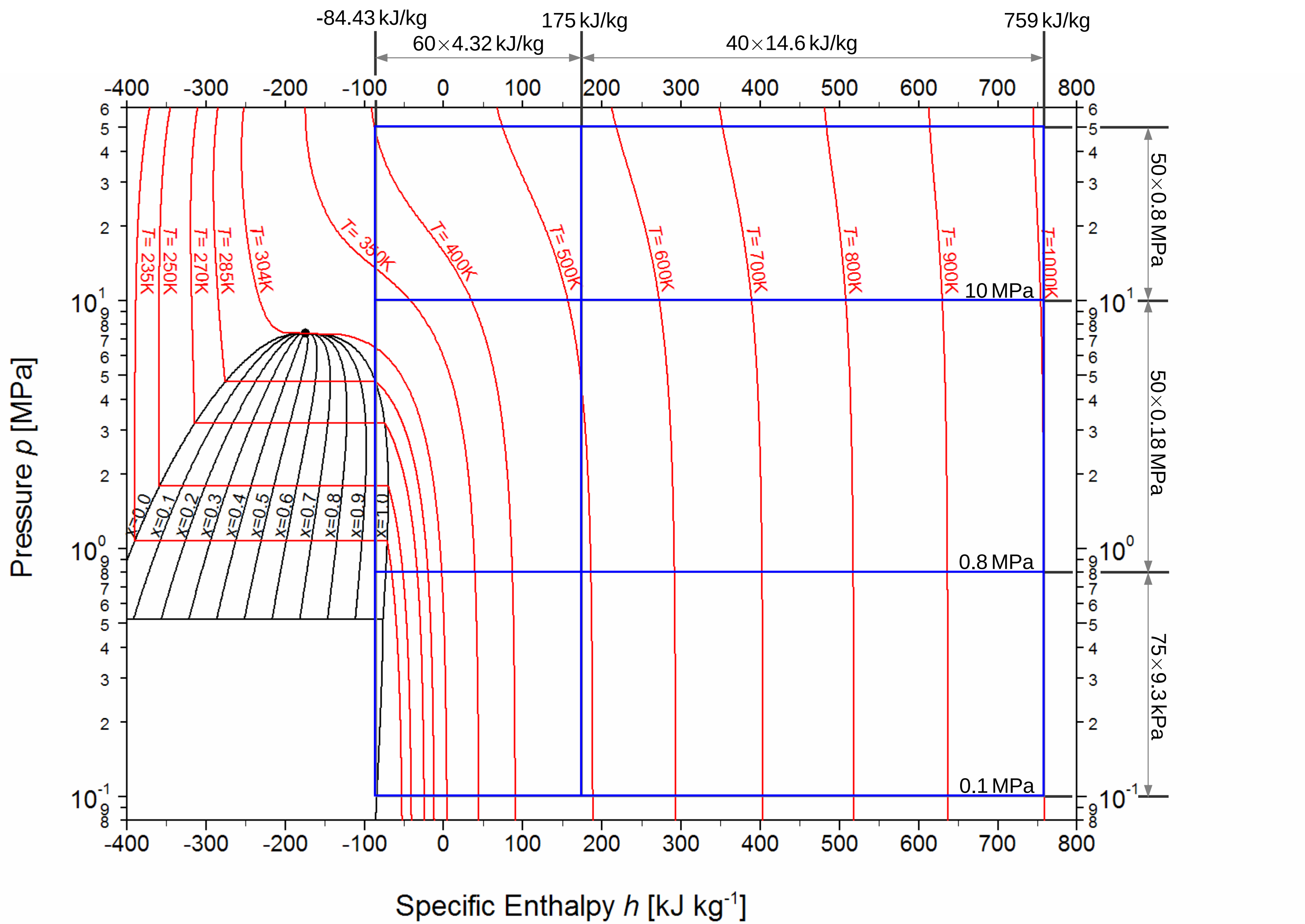
(p,T)	(p,s)	(h,s)
$h=h^{INV}(p,T)$	$h=h^{INV}(p,s)$	$p=p^{INV}(h,s)$
$s=s^{SPL}(p,h)$	$T=T^{SPL}(p,h)$	$T=T^{SPL}(p,h)$
$v=v^{SPL}(p,h)$	$v=v^{SPL}(p,h)$	$v=v^{SPL}(p,h)$
…	…	…

All thermodynamic properties, including backward functions, can be calculated without iterations.

Set of spline functions defined in a *v*-*u* grid can be created analoguously.

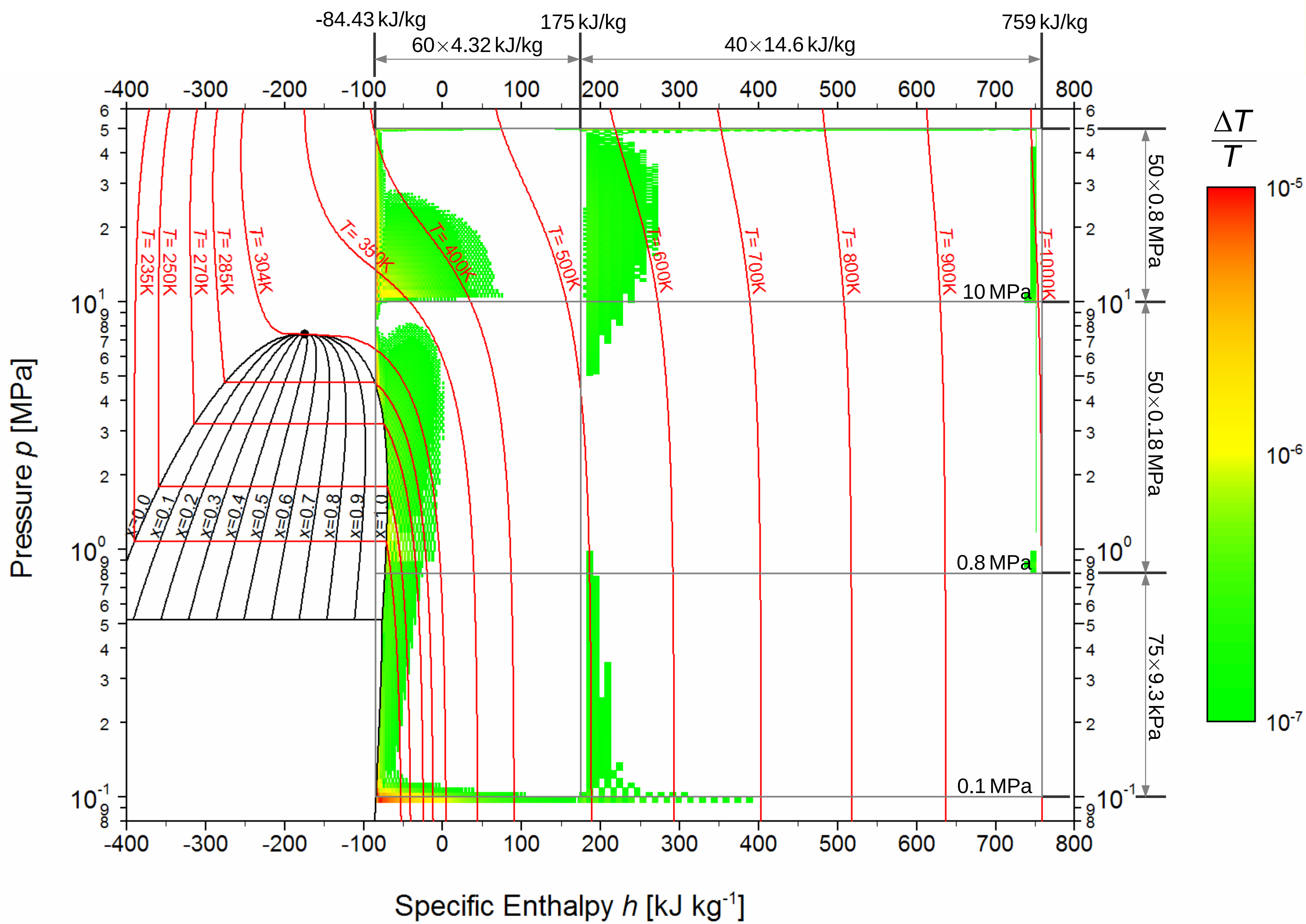
Example 1: A Set of Spline-Functions for gaseous CO_2

→ $T^{SPL}(p,h)$ and the *p*-*h* grid (calculated from scientific EOS of Span, R. and Wagner, W. (1996)):



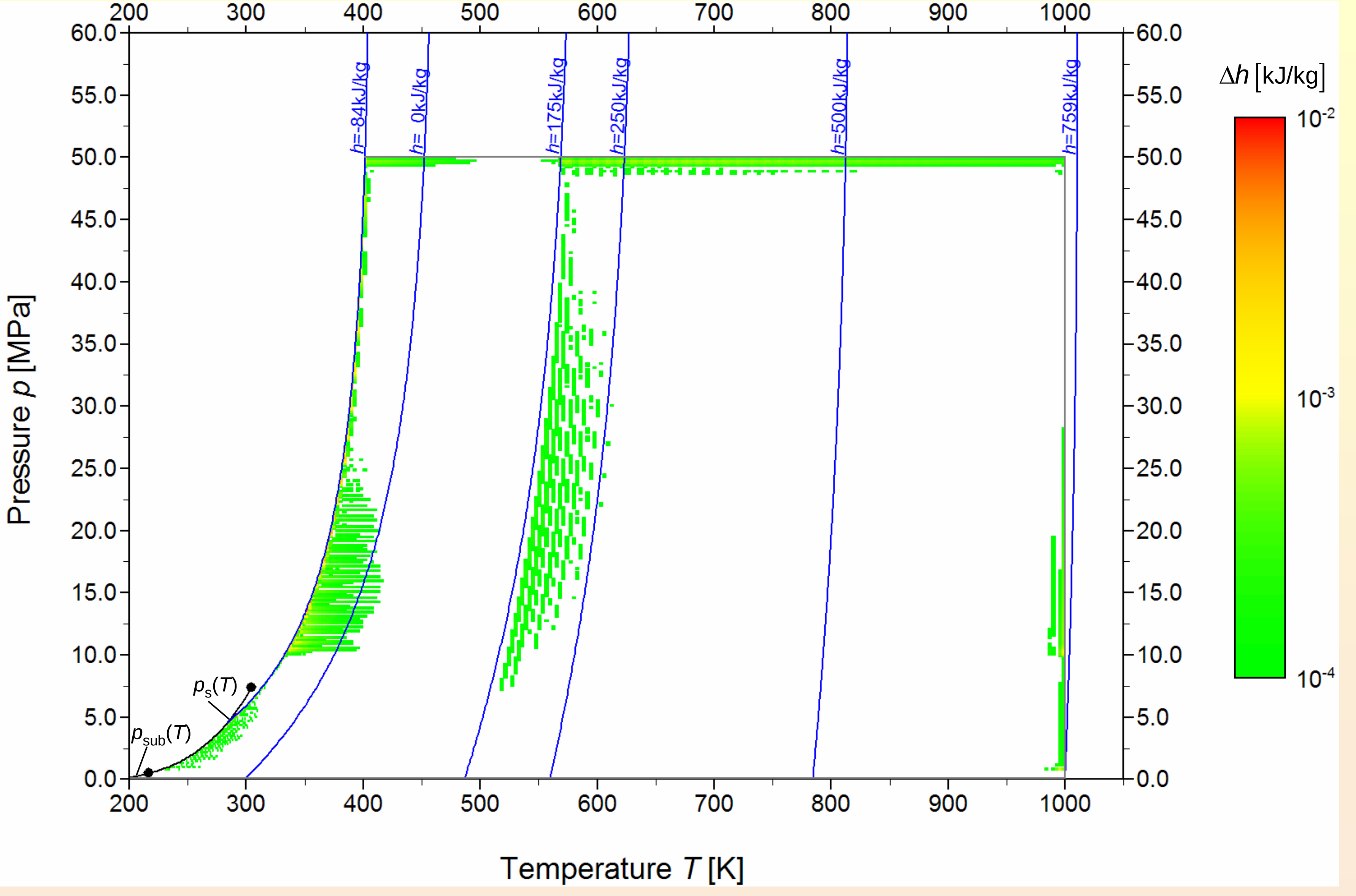
Example 1: A Set of Spline-Functions for gaseous CO_2

→ Relative deviation of $T^{SPL}(p,h)$:



Example 1: A Set of Spline-Functions for gaseous CO_2

→ Absolute deviation of $h^{INV}(p,T)$:



Computing Time Comparisons

→ Computing time has been compared with:

- EOS for scientific use of Span, R. and Wagner, W. (1996) and
- EOS for technical applications of Span, R. and Wagner, W. (2003).

} computed with REFPROP®

→ Tested on a Pentium Xeon 3.2 GHz PC with Microsoft Windows XP ®.

Computing Time Ratio (*CTR*)

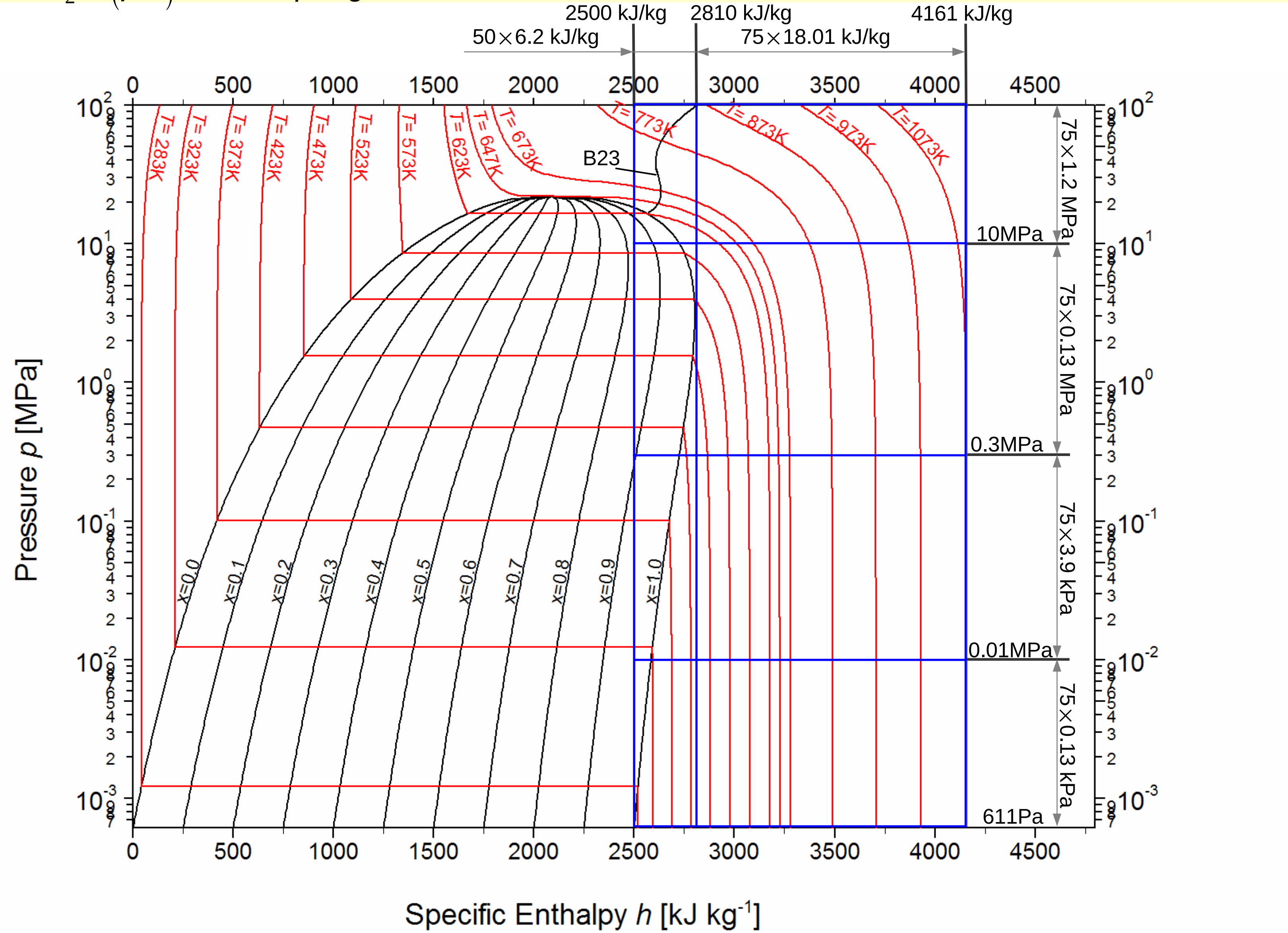
$$CTR = \frac{\text{Computing time of equation of state}}{\text{Computing time of spline function}}$$

	Spline Function		
	$T^{SPL}(p,h)$	$h^{INV}(p,T)$	$p^{INV}(h,s)$
CTR^{1996}	820	286	1250
CTR^{2003}	225	77	312

Spline-based table look-up methods are fast and accurate at the same time.

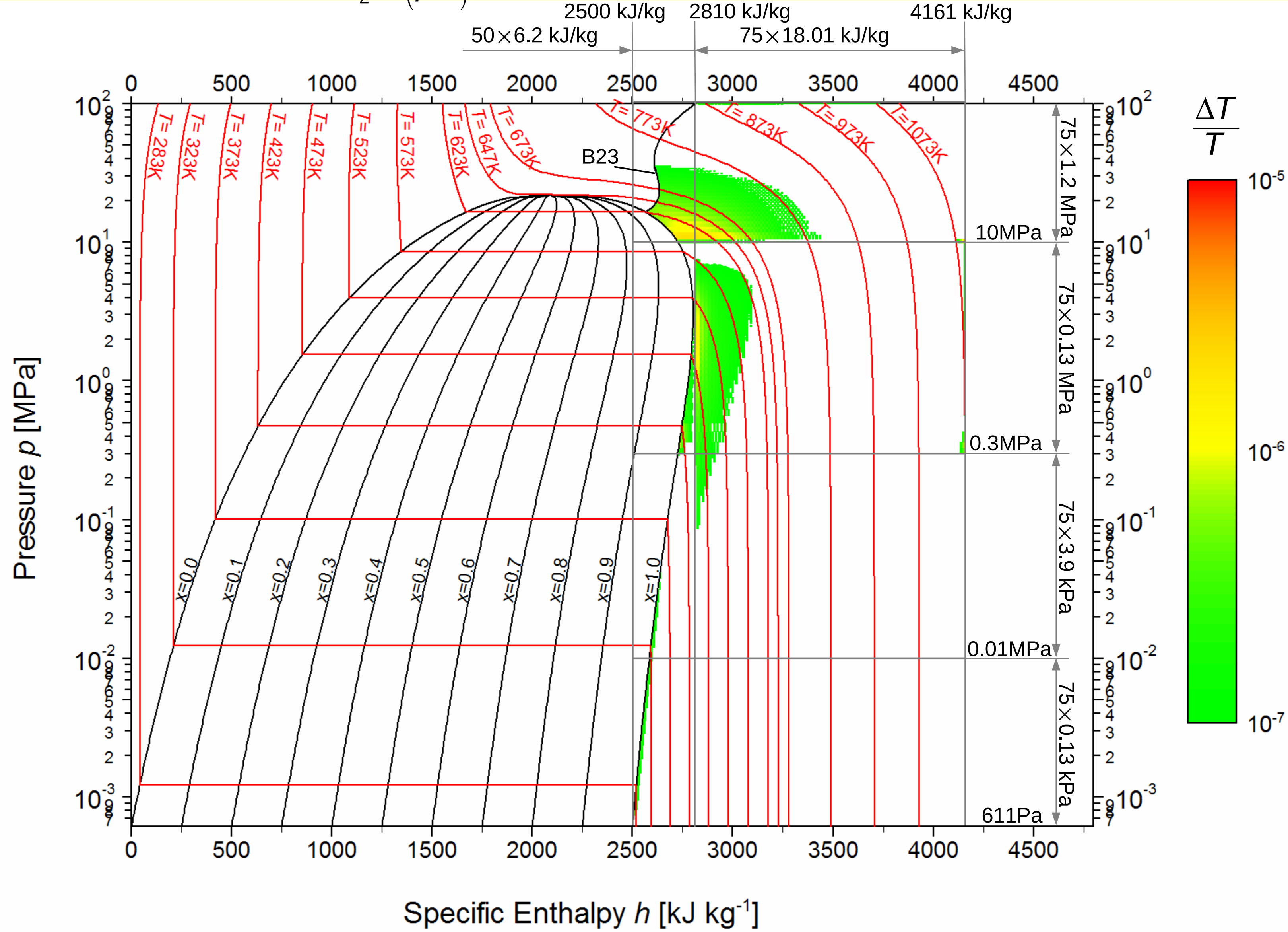
Example 2: Spline Function $T_2^{SPL}(p,h)$ and Inverse Spline Function $h_2^{INV}(p,T)$ for region 2 of IAPWS-IF97

→ $T_2^{SPL}(p,h)$ and the *p-h* grid:



Example 2: Spline Function $T_2^{\mathrm{SPL}}(p,h)$ and Inverse Spline Function $h_2^{\mathrm{INV}}(p,T)$ for region 2 of IAPWS-IF97

→ Relative deviation of $T_2^{\mathrm{SPL}}(p,h)$:



Computing Time Comparisons

- Computing time of $T_2^{\mathrm{SPL}}(p,h)$ has been compared with $T_2^{\mathrm{97BW}}(p,h)$.
- Computing time of $h_2^{\mathrm{INV}}(p,T)$ has been compared with $h_2^{\mathrm{IF97}}(p,T)$.
- Tested on a Pentium Xeon 3.2 GHz PC with Microsoft Windows XP ®.

Computing Time Ratio (*CTR*)

$$CTR = \frac{\text{Computing time of backward eq.}}{\text{Computing time of spline function}}$$

$$CTR = \frac{\text{Computing time of equation of state}}{\text{Computing time of spline function}}$$

	Spline Function	
	$T_2^{\mathrm{SPL}}(p,h)$	$h_2^{\mathrm{INV}}(p,T)$
CTR	2	1.2

Spline based table look-up methods can be faster than a backward equation.

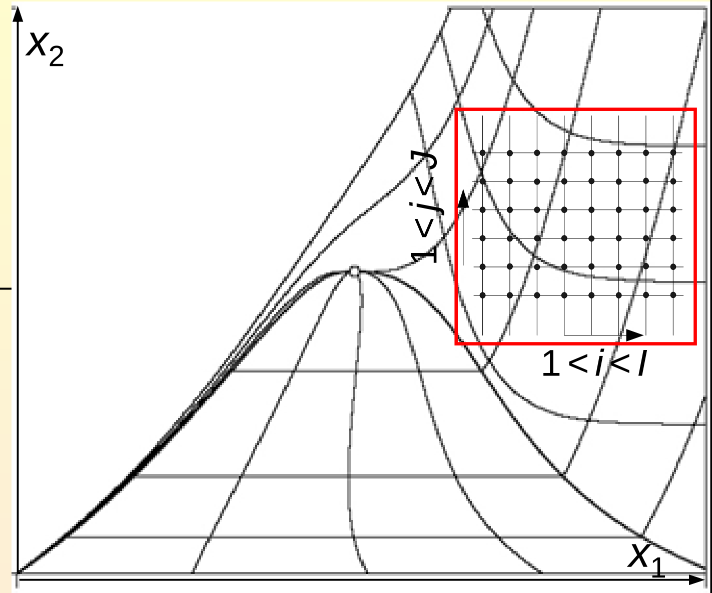
Software-Tool for Generating Spline-Based Property Libraries

Essential for the application of spline-based table look-up methods.

FluidGrid – Spline-Based Calculation of Thermodynamic Properties

INPUT

- Choice of an equation of state (pre-defined or external)
- Choice of a function
- Definition of required range of validity
- Definition of required accuracy



- Grid-optimization for desired range of validity and desired accuracy
- Calculation of all spline-coefficients
- Verification
- Estimation of CTR (computing time ratio)

OUTPUT

- Applicable source code including data grid and spline polynomials

A flexible software tool for generating spline-based property libraries is being developed.

Summary

- Spline functions are able to represent thermodynamic properties continuously.
- High accuracy and low computing times can be achieved at the same time.
- Example CO_2:
  - $T^{SPL}(p,h)$ is over 200 times faster than the iteration from the EOS $T^{EOS}(p,h)$.
  - The inverse spline function $h^{INV}(p,T)$ is over 70 times faster than the iteration from the EOS $h^{EOS}(p,T)$ and completely numerically consistent to $T^{SPL}(p,h)$.
- Example steam:
  - $T_2^{SPL}(p,h)$ is 2 times faster than the corresponding backward equation $T_2^{97BW}(p,h)$.
  - The inverse spline function $h_2^{INV}(p,T)$ is completely numerically consistent to $T_2^{SPL}(p,h)$.

Outlook

- An algorithm for grid optimization is being developed.
- The algorithm will be extended for mixtures.
- Software for automatic generation of spline functions is being developed.

Thank you for paying attention.