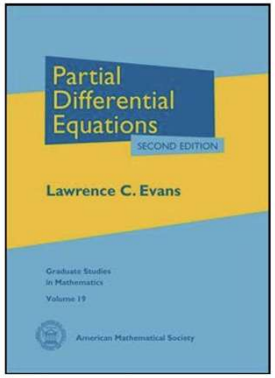
1.3 Partielle Differentialgleichungen (PDEs)

Von einer Trajektorie zu einem Feld





ODE: Ein Zustand X(t) beschreibt das System zur Zeit t

$$X'(t) = f\big(t, X(t)\big), \qquad X\colon [0,T] \to \mathbb{R}^n$$

PDE: Ein Feld y(t, x) beschreibt Zustand an jedem Ort x

$$\frac{\partial y}{\partial t} Y(t,x) = F(t,x,Y(t,x), \nabla_x Y(t,x), \Delta_x Y(t,x)), \qquad Y\colon [0,T] \times \Omega \to \mathbb{R}$$

Partielle Ableitungen



$u_t = \frac{\partial u}{\partial t}$ zeitliche Änderung bei festem Ort

$u_x = \frac{\partial u}{\partial x}$ räumliche Änderung bei fester Zeit

$u_{xx} = \frac{\partial^2 u}{\partial x^2}$ räumliche Krümmung des Profils

Nabla und Laplace Differentialoperatoren

$$\nabla Y(t,x) = grad\ Y = \left(\frac{\partial y}{\partial x_1}, \ldots, \frac{\partial y}{\partial x_n}\right)$$

$$\nabla Y(t,x) = grad\ Y = \left(\frac{\partial y}{\partial x_1}, \frac{\partial y}{\partial x_2}, \frac{\partial y}{\partial x_3}\right)$$

$$\Delta Y(t,x) = \frac{\partial^2 y}{\partial x_1^2} + \frac{\partial^2 y}{\partial x_2^2} + \frac{\partial^2 y}{\partial x_3^2}$$

$$\nabla\nabla Y(t,x) = \nabla^2 Y(t,x) = \Delta Y(t,x) = \sum_{i=1}^{n} \frac{\partial^2 y}{\partial x_i^2}$$

im dreidimensionalen kartesischen Koordinatensystem

Anfangs- und Randwertproblem



Eine PDE verknüpft eine Funktion mehrerer Variablen mit ihren partiellen Ableitungen.

Allgemeine PDE der Ordnung k

$$F\left(\frac{\partial^k z}{\partial x^k}, \ldots, \frac{\partial z}{\partial x}, z(x), x\right) = 0, \qquad x \in \Omega \subset R^{\mathrm{d}}$$

Zeitabhängige PDE mit Anfangs- und Randbedingungen

$$\frac{\partial z}{\partial t}(t,x) = F\left(\frac{\partial^k z}{\partial x^k}, (t,x), \ldots, \frac{\partial z}{\partial x}(t,x), z(t,x), q(t,x), t, x\right) \quad (t,x) \in \mathbb{R}^+ \times \Omega$$

$$z: \mathbb{R}^+ \times \Omega \to \mathbb{R},$$

Anfangsbedingung: Zustand bei $t = 0$

$$z(0,x) = z_0(x), \qquad x \in \Omega$$

Gibt das gesamte räumliche Anfangsprofil vor.

Randbedingungen: Vorgaben auf ∂Ω

Dirichlet: $z(t,x) = g(t,x)$

Neumann: $\partial_{\mathrm{n}} z(t,x) = q(t,x)$

Wert oder Normalableitung am Rand, für t > 0.

Lineare partielle Differentialgleichungen



Lineare PDE

$$\sum_{|\alpha|\leq k} a_\alpha(x)\,\frac{\partial^\alpha y}{\partial x^\alpha} + a_0(x), \quad x \in U \subseteq \mathbb{R}^n,$$

Quasi lineare PDE

$$\sum_{|\alpha|=k} a_\alpha(x, y, \frac{\partial y}{\partial x}, \ldots, \frac{\partial^{k-1} y}{\partial x^{k-1}})\,\frac{\partial^\alpha y}{\partial x^\alpha} + a_0(x, y, \frac{\partial y}{\partial x}, \ldots, \frac{\partial^{k-1} y}{\partial x^{k-1}})$$

Eine PDE ist quasi-linear, wenn sie in den Ableitungen höchster Ordnung linear ist, kann jedoch in Termen niedriger Ordnung nichtlinear sein

Homogene und inhomogene PDEs



Homogene PDE

$$G\left(x, y(x), \frac{\partial y}{\partial x}, \ldots, \frac{\partial^k y}{\partial x^k}\right) = 0.$$

Inhomogene PDEs

$$G(x, y(x), \frac{\partial y}{\partial x}, \ldots, \frac{\partial^k y}{\partial x^k}) = f(x), \quad f(x) \neq 0.$$

Eine PDE ist quasi-linear, wenn sie in den Ableitungen höchster Ordnung linear ist, kann jedoch in Termen niedriger Ordnung nichtlinear sein

Parabolische partielle Differentialgleichungen

Allgemeine Form einer linearen parabolischen PDE in einer Raumdimension

$$\begin{cases} \frac{\partial z}{\partial t}(t,x) = a(t,x)\frac{\partial^2 z}{\partial x^2}(t,x) + b(t,x)\frac{\partial z}{\partial x}(t,x) + c(t,x)z(t,x) + f(t,x), & (t,x) \in \mathbb{R}^+ \times [0,1], \\ z(0,x) = z_0(x), & x \in [0,1], \\ z(t,x) = z_\Gamma(x), & t,x \in \mathbb{R}^+ \times \{0,1\}. \end{cases}$$

Parabolische Bedingung

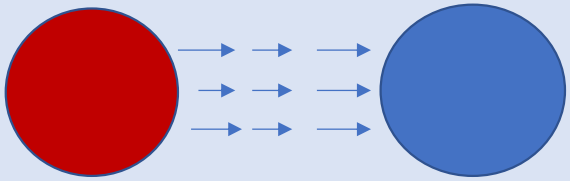
Die PDE ist parabolisch, wenn $a > 0$ für alle x

Wärmeleitung:

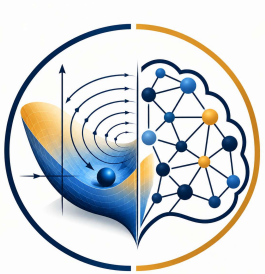
$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}, \alpha > 0$$

Diffusionsgleichung

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2}, D > 0$$



Hyperbolische partielle Differentialgleichungen



Allgemeine Form einer skalaren Transportgleichung 1. Ordnung

$$\begin{cases} \dfrac{\partial z}{\partial t}(t,x) = F\left(\dfrac{\partial z}{\partial x}(t,x), z(t,x), x\right), & (t,x) \in \mathbb{R}^+ \times U, \\ z(0,x) = z_0(x), & x \in U, \\ z(t,x) = z_\Gamma(x), & t,x \in \mathbb{R}^+ \times \Gamma. \end{cases}$$

Lineare hyperbolische PDEs

Die PDE ist linear hyperbolisch, sobald F linear hinsichtlich der ersten beiden Argumente ist

Transportgleichung für Schadstoffausbreitung:

$$\frac{\partial c}{\partial t} = v\frac{\partial c}{\partial x}, \qquad v > 0$$

Konvektions-Diffusionsgleichung mit Quellterm



$$\frac{\partial c}{\partial t}(t,x) + v\frac{\partial c}{\partial x}(t,x) = D\frac{\partial^2 c}{\partial x^2}(t,x) + q(t,x), t > 0, x \in [0,L],$$

$$c(0,x) = c_0(x), x \in [0,L],$$
$$c(t,0) = c_L(t), t > 0$$
$$c(t,L) = c_R(t), \mathrm{t} > 0$$

Konvektions-Diffusionsgleichung mit Quellterm



$$c_t + 5c_x = 0.1c_{xx} + q(t,x), t > 0, x \in (0,100),$$

$$c(0,x) = c_0(x), x \in [0,L],$$
$$c(t,0) = 0$$
$$c(t,100) = 0$$

Elliptische Partielle Differentialgleichungen



Allgemeine Form eine

$$\begin{cases} -\sum_{i,j=1}^{n} a_{ij}(x) u_{x_i x_j} + \sum_{i=1}^{n} b_i(x) u_{x_i} + c(x)u = f(x), & x \in \Omega, \\ u(x) = g(x), & x \in \partial\Omega. \end{cases}$$

Elliptische PDE

Stationäre räumliche Verteilungen

Poisson-Gleichung:

$$-\left(\frac{\partial^2 u}{\partial x^2}\right) = f(x)$$

Klassifikation von PDEs



Allgemein: Hauptteilmatrix

$$-\sum_{i,j=1}^{k} a_{ij}(x)\partial_i\partial_j u(x) + \sum_{i=1}^{k} b_i(x)\partial_i u(x) + c(x)u(x) = f(x),$$

$$A(x) = (a_{ij}(x)) \in \mathbb{R}^{k\times k}$$

Klassifikation über die Eigenwerte von A, eine lineare PDE 2. Ordnung heißt:

Elliptisch: A(x) positiv definit, alle Eigenwerte positiv (oder negativ)
Hyperbolisch: genau ein abweichendes Vorzeichen
Parabolisch: positiv semidefinit (alle Eigenwerte positiv oder negativ), aber nicht positiv definit (mindestens ein Null-Eigenwert)

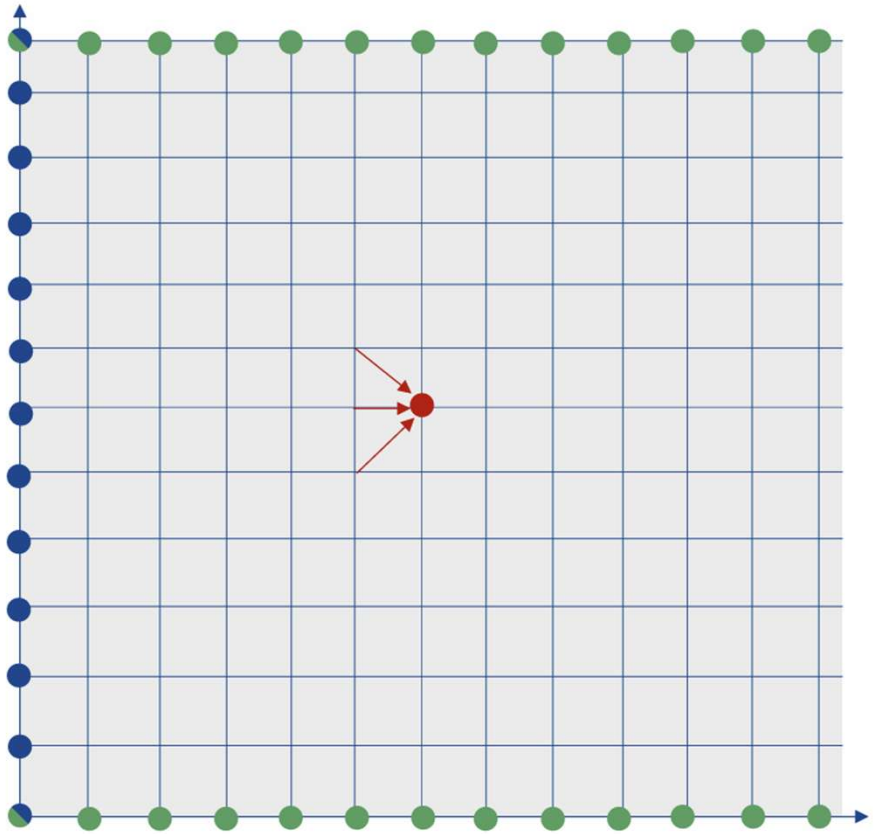
Spezialfall: zwei Variablen

$$a\partial_1^2 u + 2b\partial_1\partial_2 u + c\partial_2^2 u = f(x_1, x_2, u, \partial_1 u, \partial_2 u).$$

$$ac - b^2 \begin{cases} < 0, & \textit{hyperbolische PDE}, \\ > 0, & \textit{elliptische PDE}, \\ = 0, & \textit{parabolische PDE}. \end{cases}$$

1.4 Numerische Schemata für PDEs - FDM (I)

Idee: Ableitungen einer PDE durch Differenzenquotienten auf einem Gitter approximieren



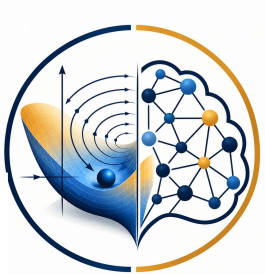
Finite Differenzen Methode

Diskretisierung der Ableitungen

$$x_i = i\delta_x$$
$$t_n = n\delta_t$$

$$u_t(t,x) \approx \frac{u(t+\delta_t,x) - u(t,x)}{\delta_t}$$
$$u_x(t,x) \approx \frac{u(t,x) - u(t,x-\delta_x)}{\delta_x}$$
$$u_{xx}(t,x) \approx \frac{u(t,x+\delta_x) - 2u(t,x) + u(t,x-\delta_x)}{\delta_x^2}$$

1.4 Numerische Schemata für PDEs – Galerkin Methode (II)

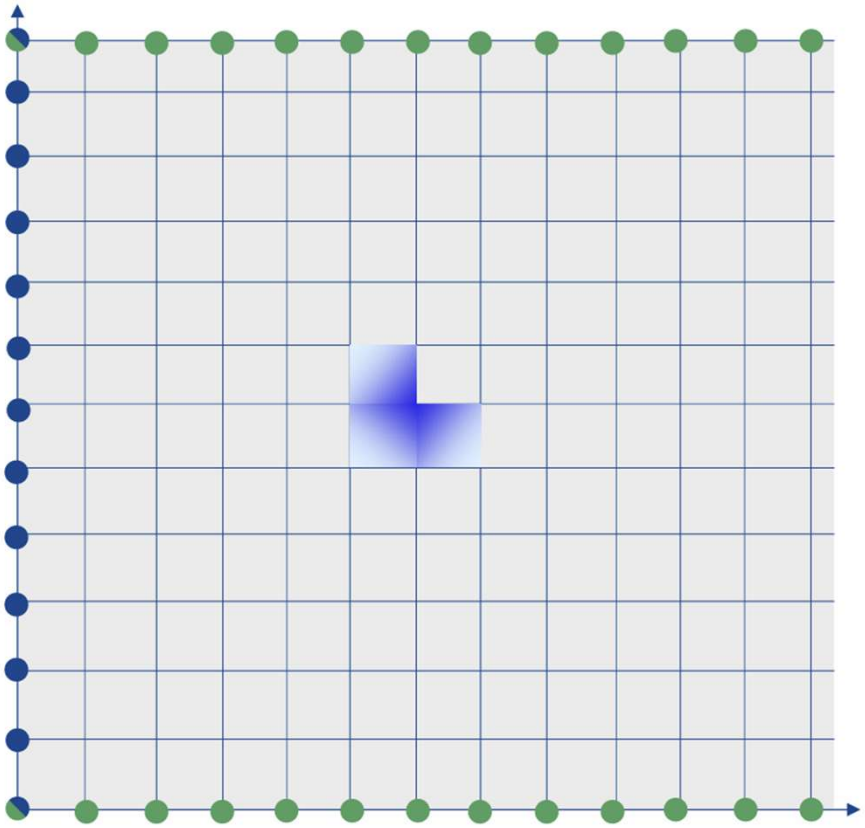


System

$$\frac{d^2u}{dx^2}(x) = -f(x), \quad x \in (0,1)$$

$$u(0) = u(1) = 0 \qquad f(x) = 1$$

1.4 Numerische Schemata für PDEs - FEM (III)



Finite Elemente Methode

1. Sie unterteilen Ihren Raum in einzelne Elemente $\{E_i\}$

2. Man wählt für jedes Element eine Basis aus,
   z. B. affine Funktionen $\phi_i(t,x) = at + bx + c$

3. Die approximierte Lösung ist
   $$u(t,x) \cong \phi_i(t,x) \text{ if } (t,x) \in E_i$$

4. Die schwache Formulierung sollte für jedes Element gelten

1.4 Numerische Schemata für PDEs – FEM (III)



System

$$\frac{d^2u}{dx^2}(x) = -f(x), \quad x \in (0,1)$$

$$u(0) = u(1) = 0 \qquad f(x) = 1$$

Schwache Formulierung

$$C_c^\infty = \{\phi \in H^1 \mid \phi(0) = \phi(1) = 0\}$$

$$\int_0^1 u'(x)\phi'(x)dx = \int_0^1 f(x)\phi(x)dx$$

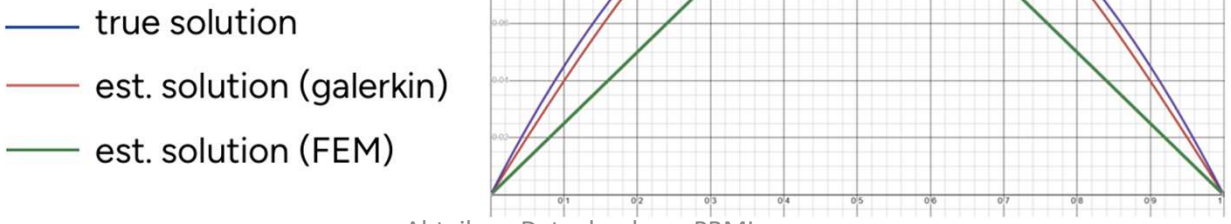
Basis Funktionen

$$\phi_1(x) = x$$
$$\phi_2(x) = 1 - x$$

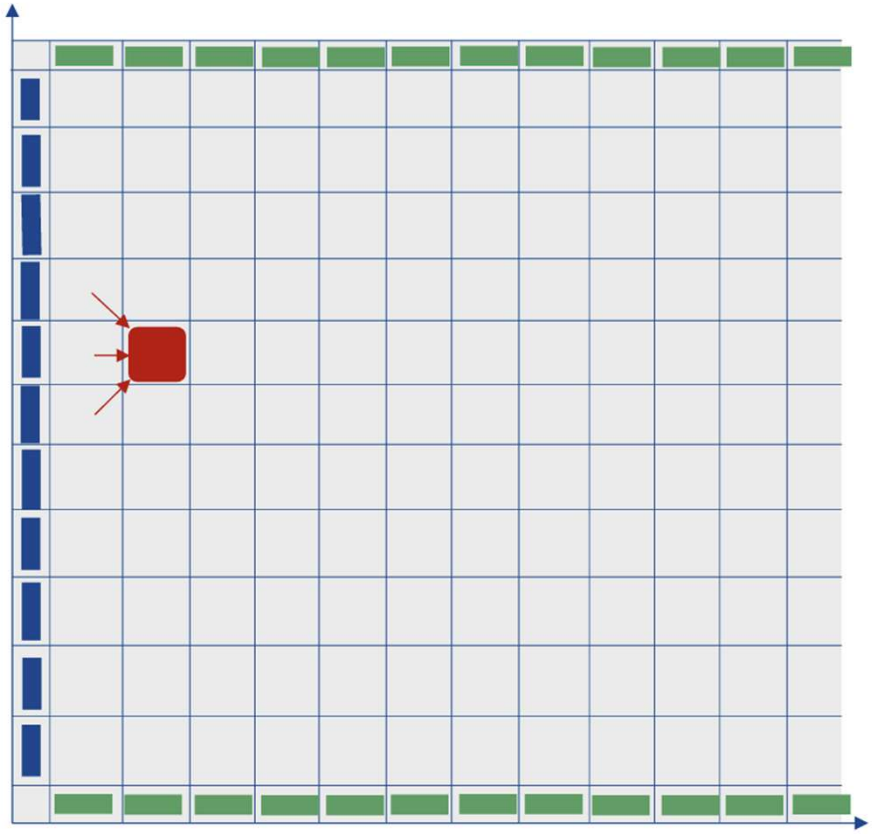
$$u(x) = \begin{cases} c\phi_1(x) \text{ if } x \in [0, \frac{1}{2}], \\ c\phi_2(x) \text{ if } x \in [\frac{1}{2}, 1] \end{cases}$$

$$c\left(\int_0^{0.5} \phi_1'(x)^2 dx + \int_{0.5}^1 \phi_2'(x)^2 dx\right) = 2\int_0^{0.5} \phi_1(x)dx$$

$$c = 0.25$$



1.4 Numerische Schemata für PDEs – FVM (IV)



Finite Volumen Methode

1. Sie unterteilen Ihren Raum in Volumen $\{V_i\}$
2. Die approximierte Lösung ist in jedem Volumen konstant
3. Wir berechnen das Integral über jedes Volumen der PDE
4. Wir approximieren Ableitungen mithilfe finiter Differenzen
5. Wir berechnen den konstanten Wert für jedes Volumen

1.4 Numerische Schemata für PDEs - Zusammenfassung



Merkmal	Galerkin-Verfahren	Finite-Elemente-Methode (FEM)	Finite-Volumen-Methode (FVM)	Finite-Differenzen-Methode (FDM)
Typ	Variationsprojektion	Galerkin + lokale Basis auf einem Netz	Integrale Erhaltung	Punktweise Approximation von Ableitungen
Grundidee	Residuum orthogonal zu Testfunktionen	Wie Galerkin, mit stückweise polynomialer Basis	Flussbilanz über Kontrollvolumen	Ableitungen durch Differenzenquotienten ersetzen
Testfunktionen	Erforderlich (wie die Basis, oft global)	Lokale Basis-/Testfunktionen (z. B. Hutfunktionen)	Keine explizit – flussbasiert	Nicht verwendet
Basisfunktionen	Beliebig (global oder lokal)	Stückweise polynomial, kompakter Träger	Keine – Lösung als Mittelwerte pro Zelle	Nicht anwendbar
Formulierung	Schwach (integral/variationell)	Schwach (Galerkin-FEM)	Integrale Erhaltungsform	Stark (Differentialform)
Geometrische Flexibilität	Begrenzt	Sehr gut (komplexe/ gekrümmte Geometrien)	Sehr gut (insbesondere in der CFD)	Gering (meist strukturierte Gitter)
Matrixstruktur	Vollbesetzt (bei globaler Basis)	Dünnbesetzt (lokale Basisfunktionen)	Dünnbesetzt (lokale Flusskopplung)	Dünnbesetzt (lokale Differenzenstencils)
Implementierungsaufwand	Mittel	Höher (Netz, Formfunktionen, Assemblierung)	Mittel bis hoch (Netz + Flusslogik)	Sehr gering (insbesondere bei regulären 1D-/2D-Gittern)
Anwendungen	Spektralmethoden, Theorie	Strukturmechanik, elliptische PDEs, Festkörpermodellierung	CFD, Massen-/Energieerhaltung, poröse Medien	Schnelle Prototypen, einfache PDEs, einfache Geometrien



Anwendungsbeispiel 3D Druck

Formulierung eines Anfangs-Randwertproblems für selektives Laserschmelzen



Wärmeleitungsgleichung

$$c\rho\frac{\partial}{\partial t}T=\kappa\left(\frac{\partial^2}{\partial x^2}T+\frac{\partial^2}{\partial y^2}T+\frac{\partial^2}{\partial z^2}T\right),\quad (x,y,z)\in\Omega,\quad t\in[0,T]$$

$$\frac{\partial}{\partial t}T=\alpha\,\Delta T$$

$$T(t_0,x,y,z)=T_0,\quad (x,y,z)\in\Omega$$

$$T(t,x,y,z)=T_{\mathcal{B}},\quad (x,y,z)\in\Omega,\quad t\in[0,T]$$

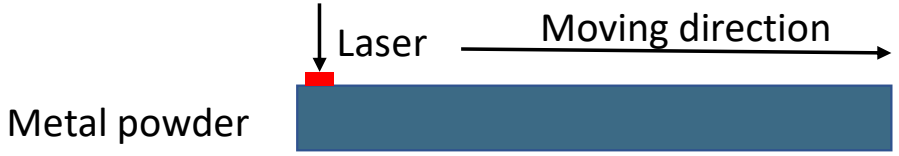
$$\kappa\frac{\partial T}{\partial n}=q,\quad (x,y)\in\Omega_S$$

$$q=\frac{2Ap}{\pi r_b^2}\exp\left(\frac{2(\mathbf{r}_0-\mathbf{v}t)^2}{r_b^2}\right)$$

with $\mathbf{r_0}=(\frac{x_b}{4},\frac{y_b}{2})$

Anfangs-und Randbedingungen

Moving heat flux



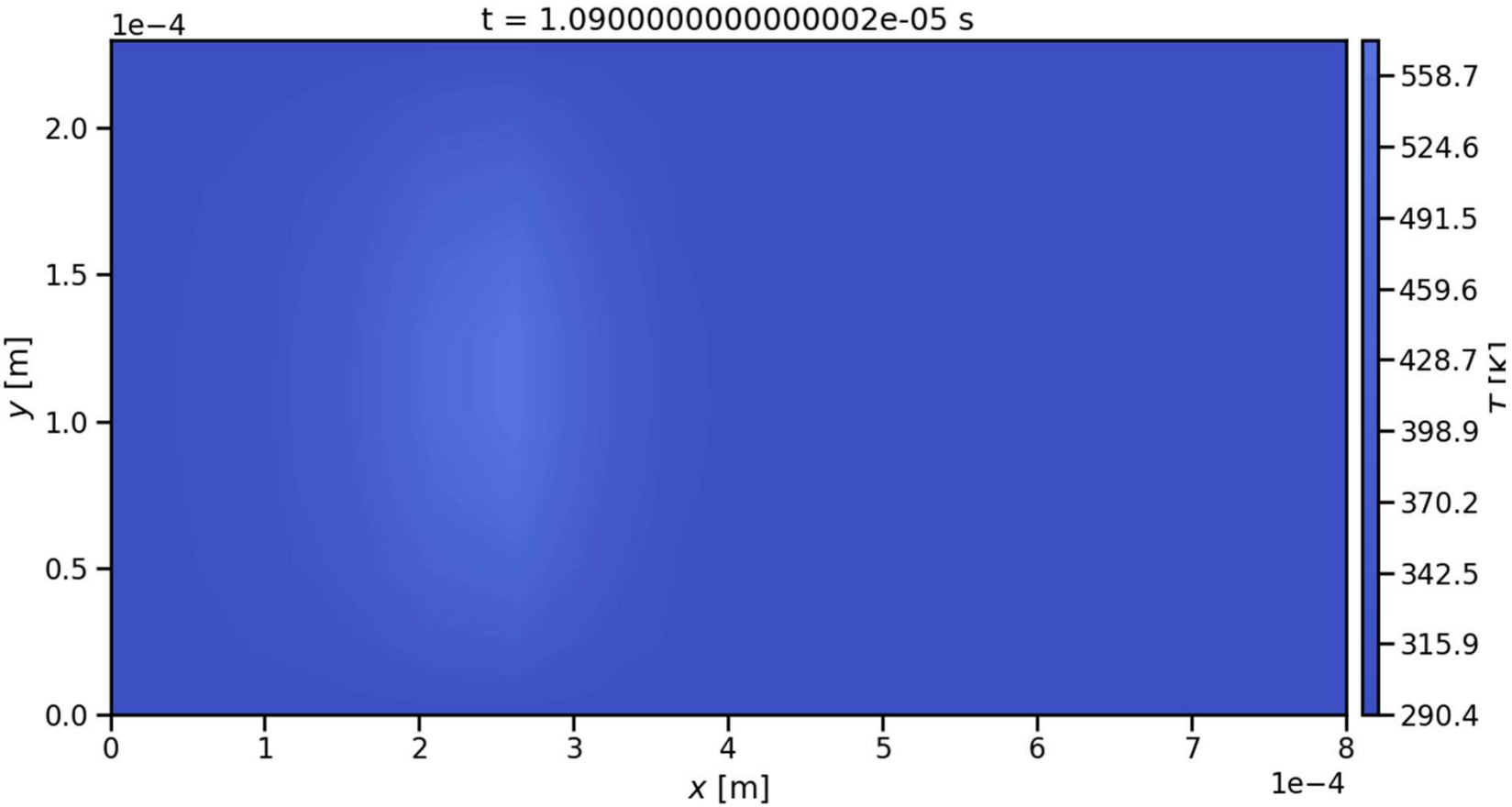
Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.

Material und Prozessparameter

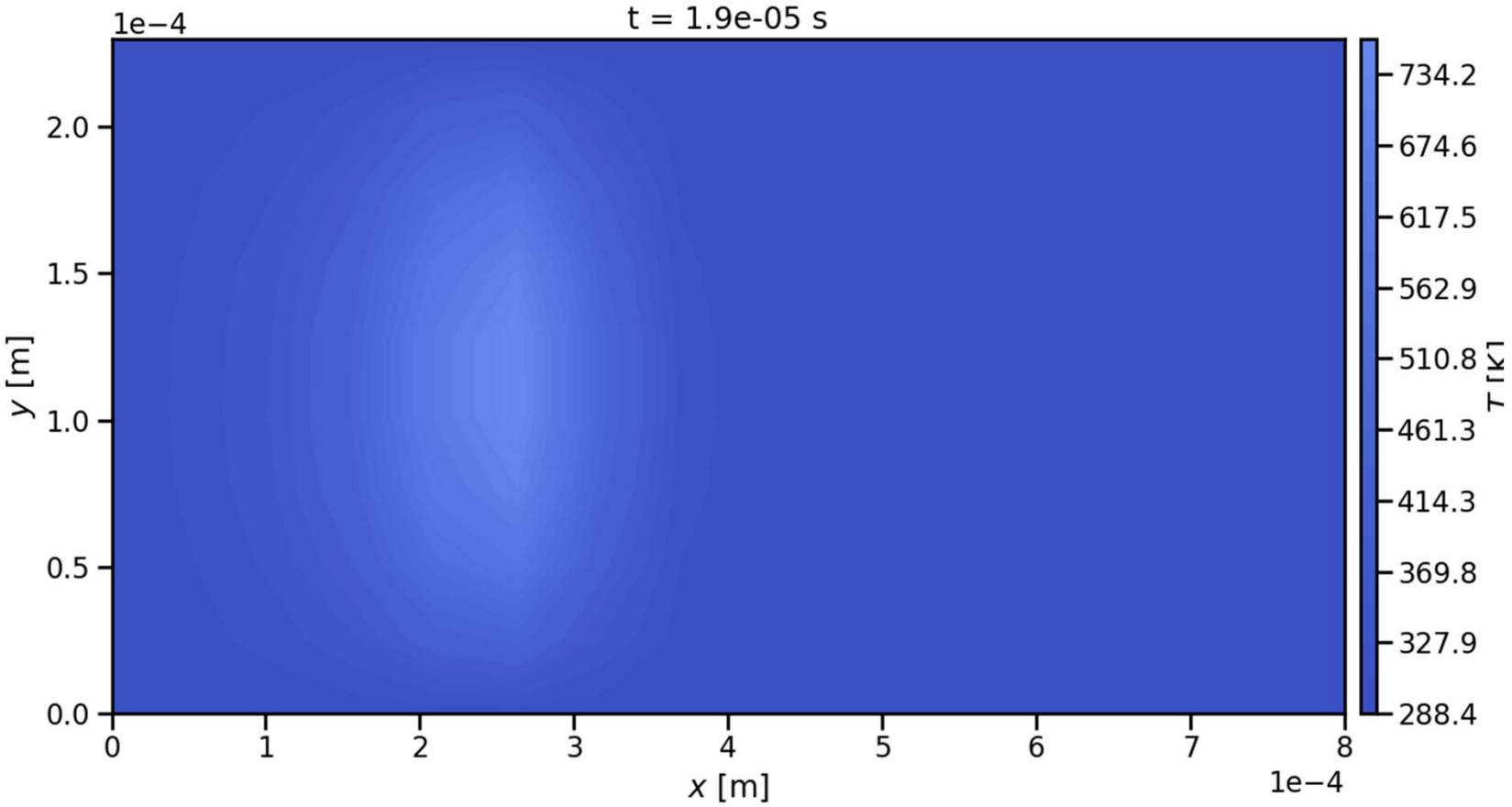


Properties	
Laser Power P(W)	250
Ambient Temperature T_0 (K)	293.15
Density ρ (kg/m^3)	$0.6 \cdot 7800$
Thermal conductivity κ (W/(m K))	$0.6(11.82 + 1.06 \times 10^{-2}\,\mathrm{T})$
Specific heat c (J/(kg K))	$330.9 + 0.563T - 4.015 \times 10^{-4}\,\mathrm{T}^2$ $+9.465 \times 10^{-8}\,\mathrm{T}^3$
Absorption of the laser energy A	0.09
Laser Beam Radius r(m)	1.4×10^{-4}
Latent Heat of Fusion H (J/kg)	272×10^{-3}

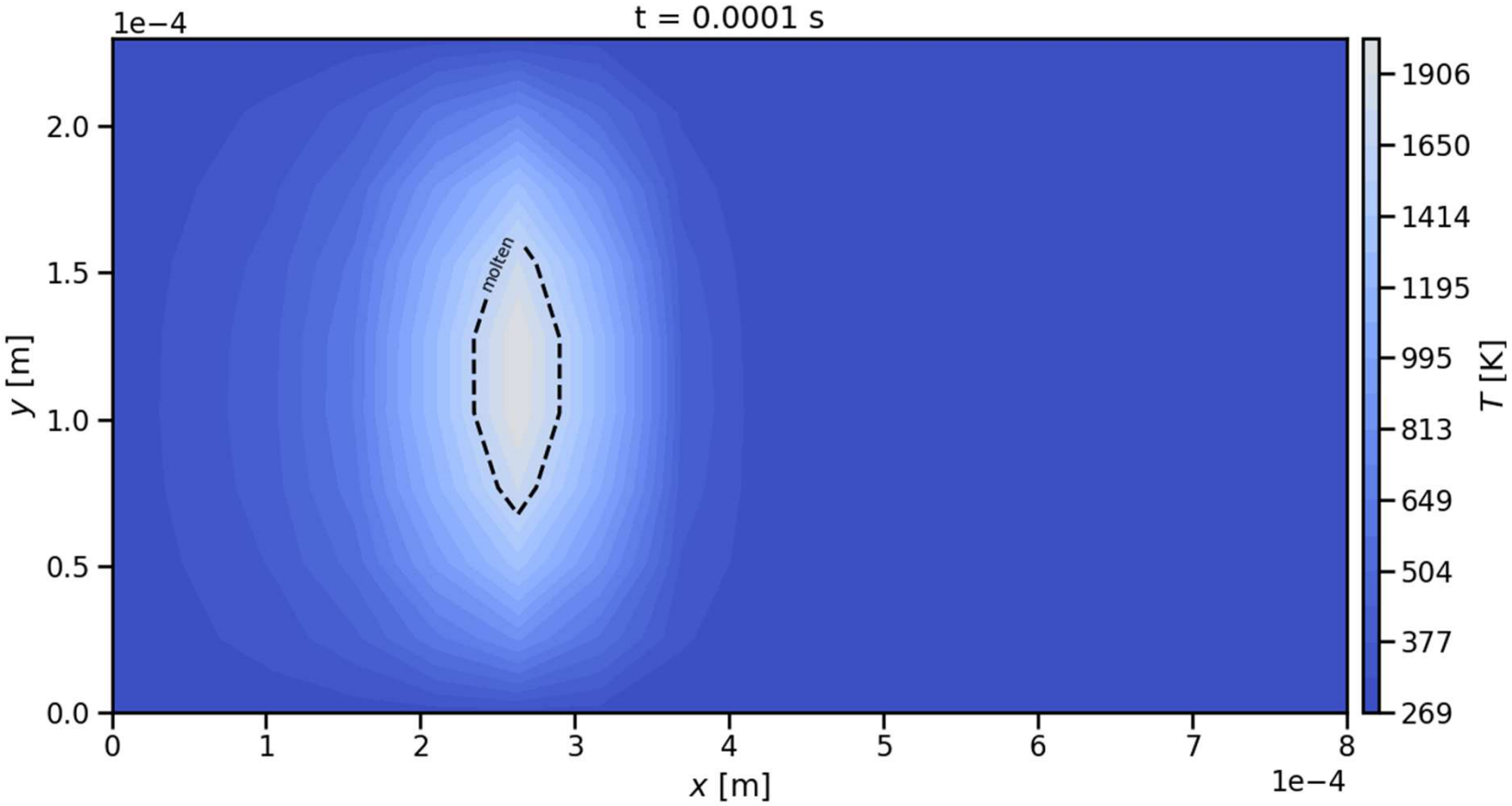
Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.



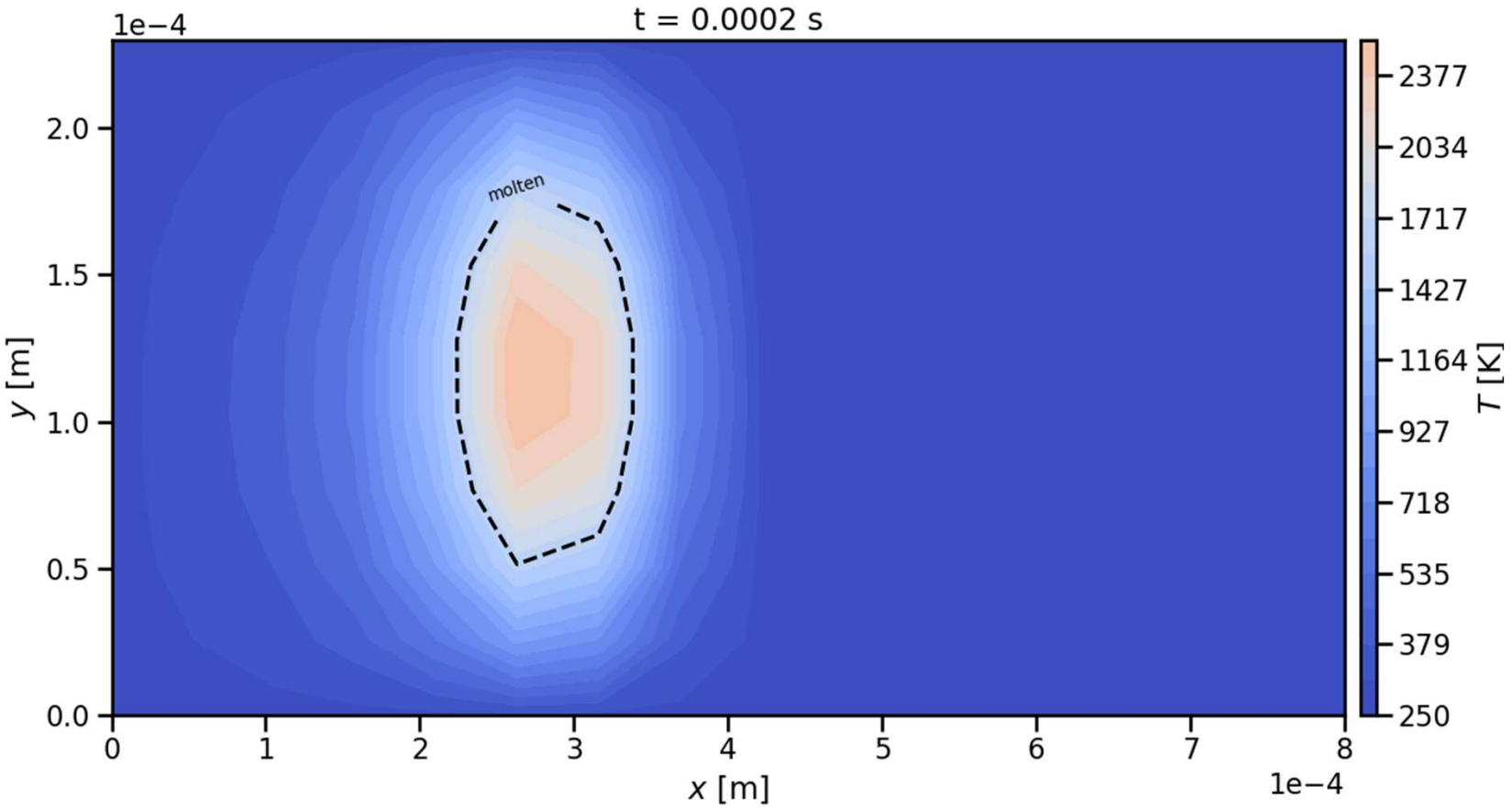
Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.



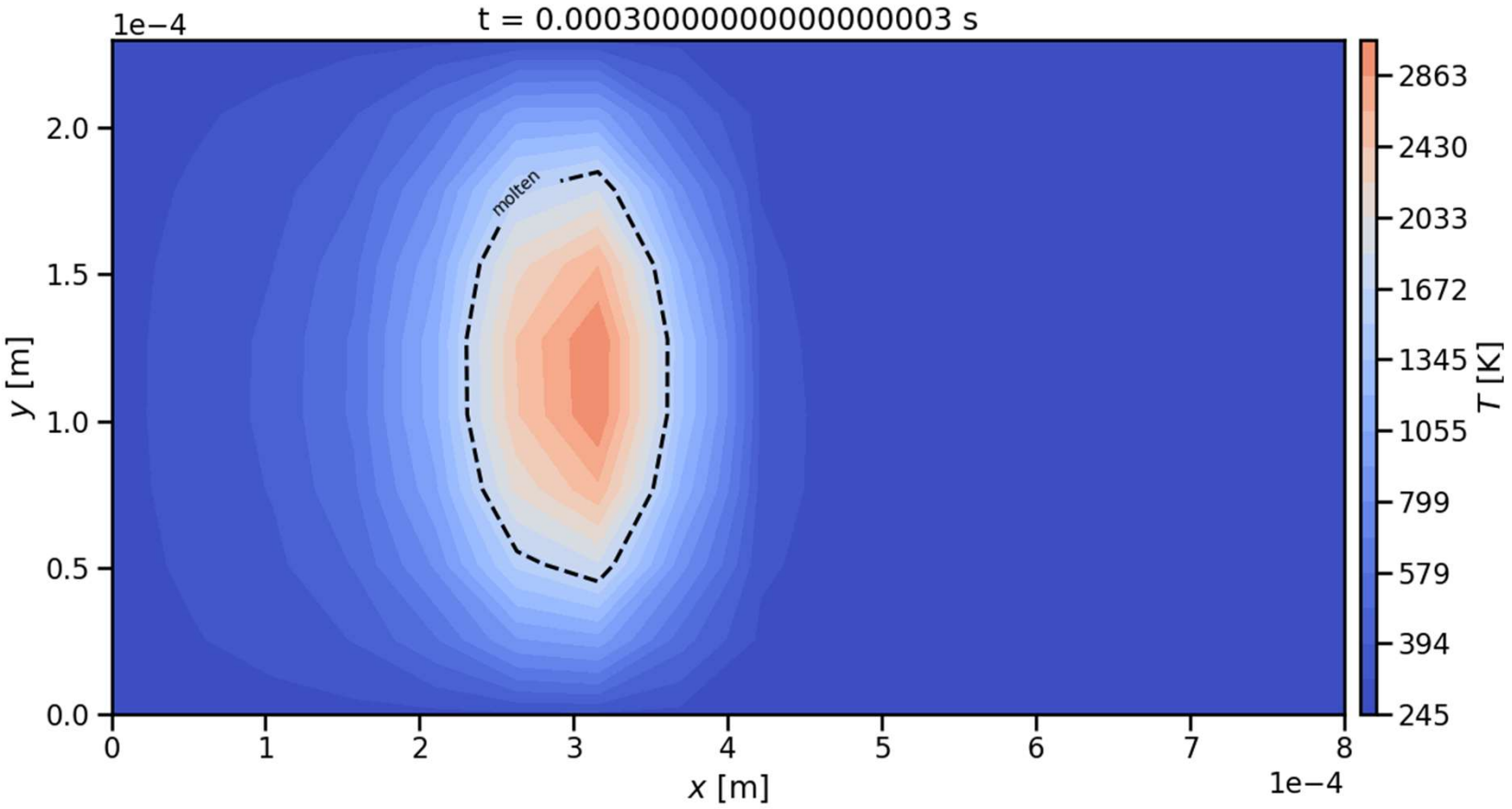
Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.



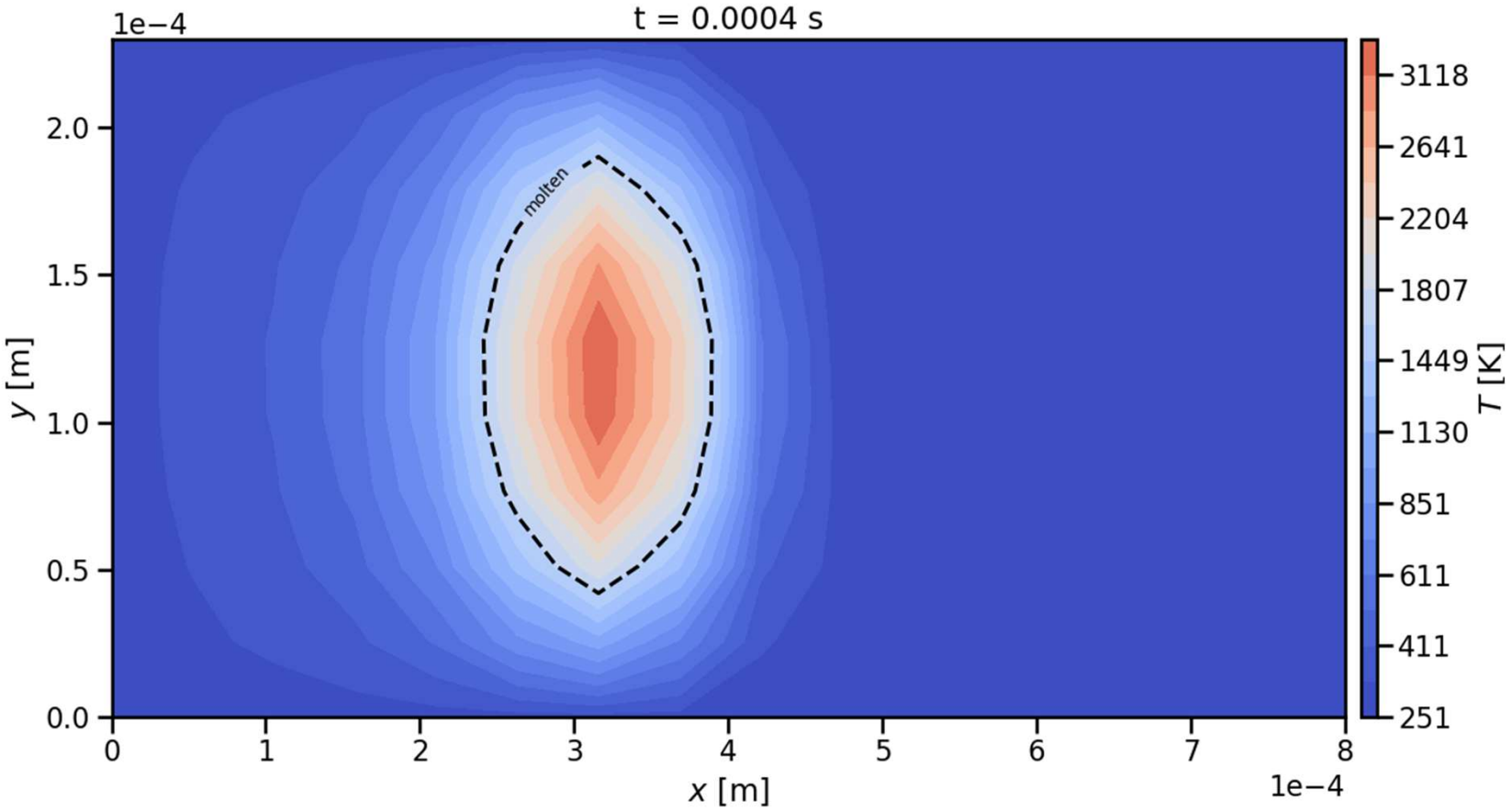
Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.



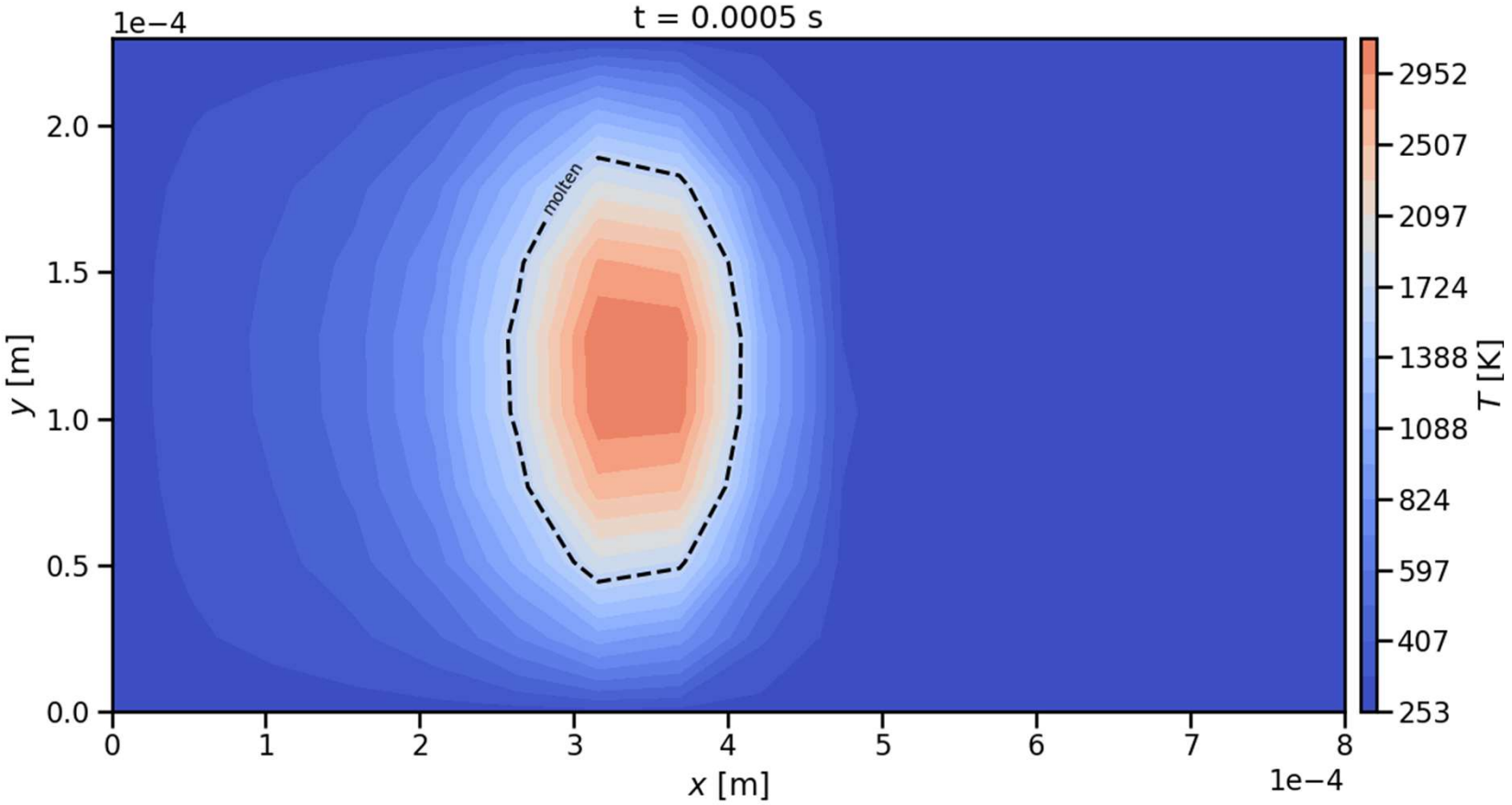
Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.



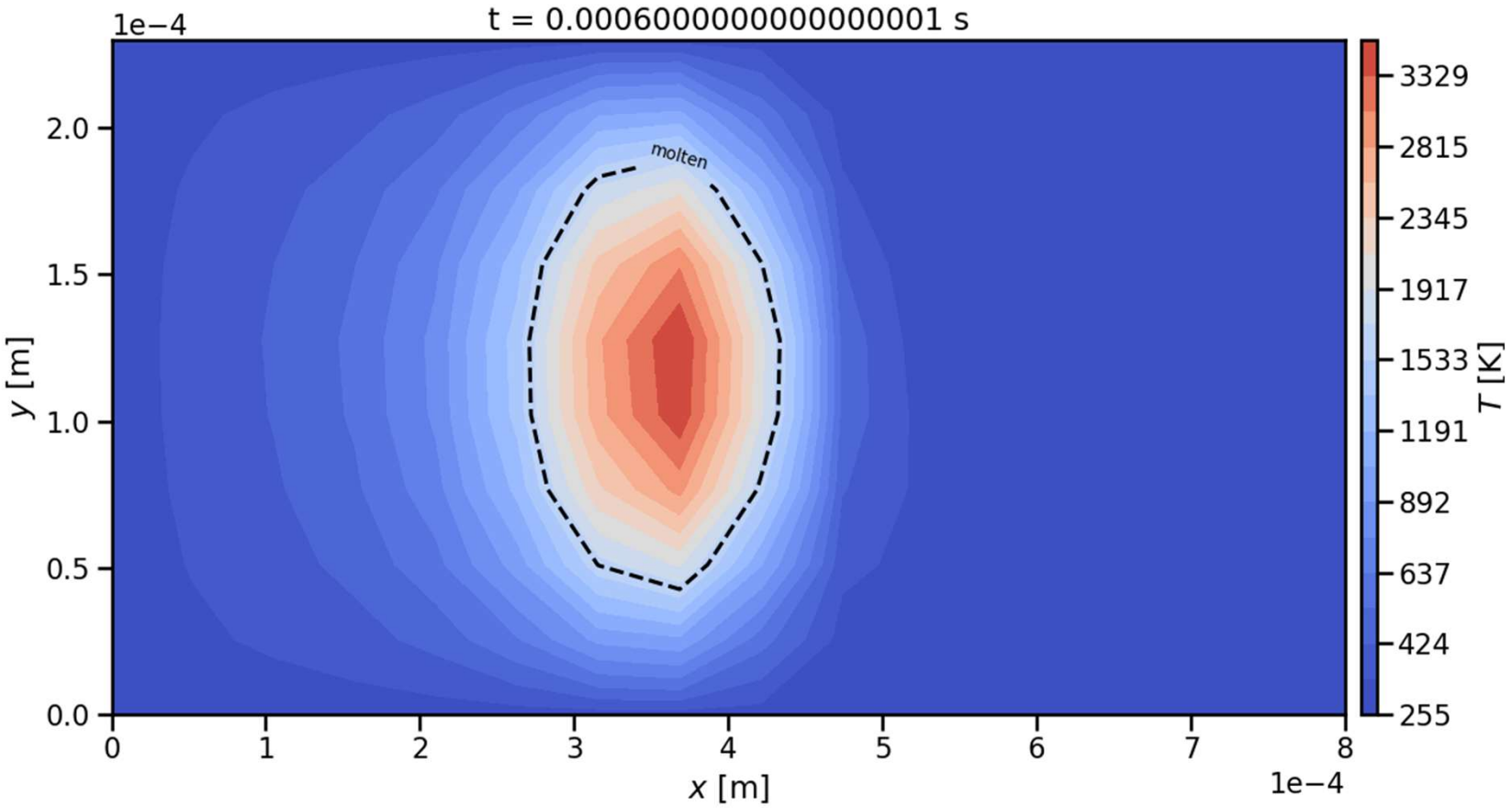
Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.



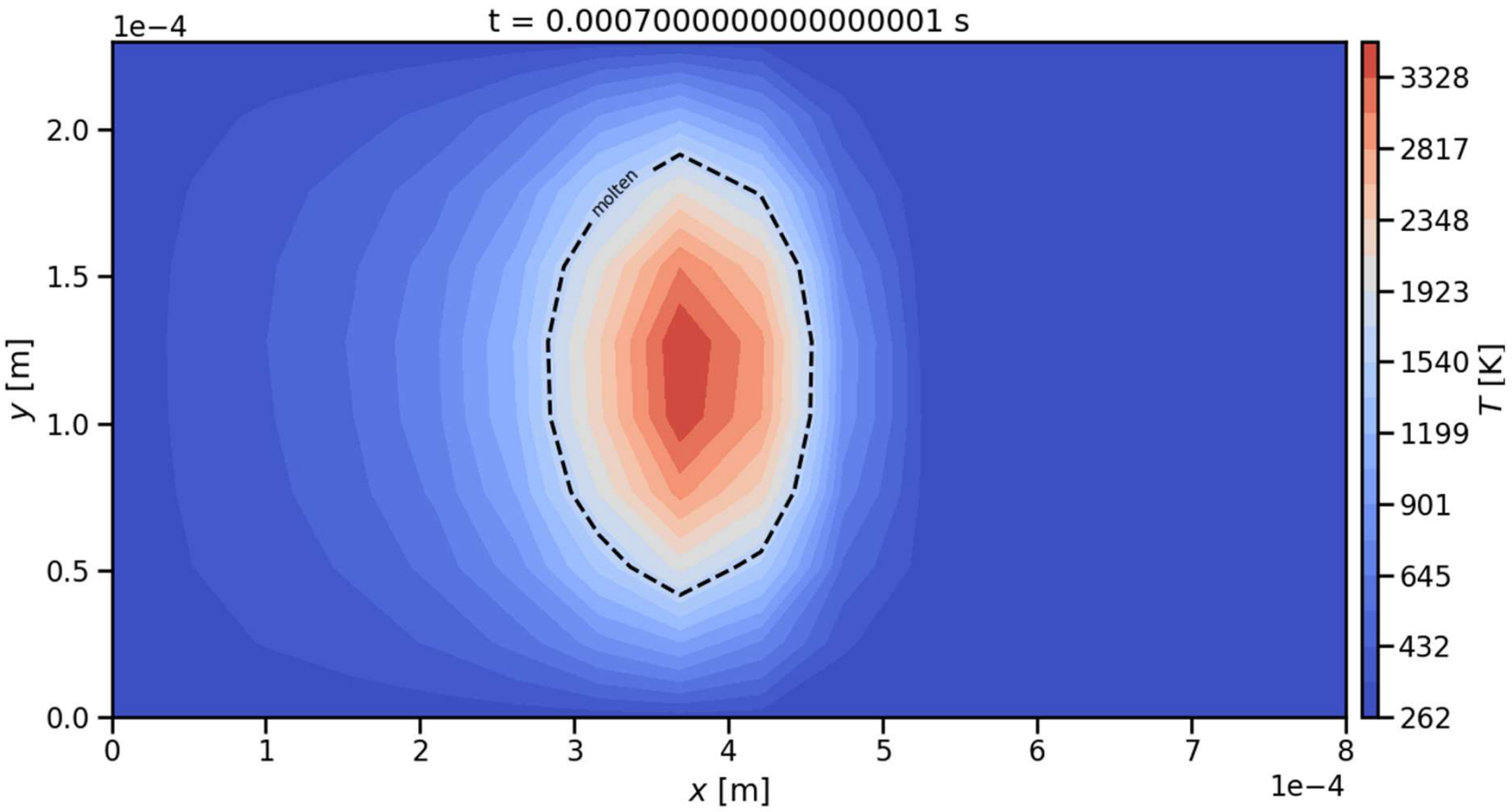
Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.



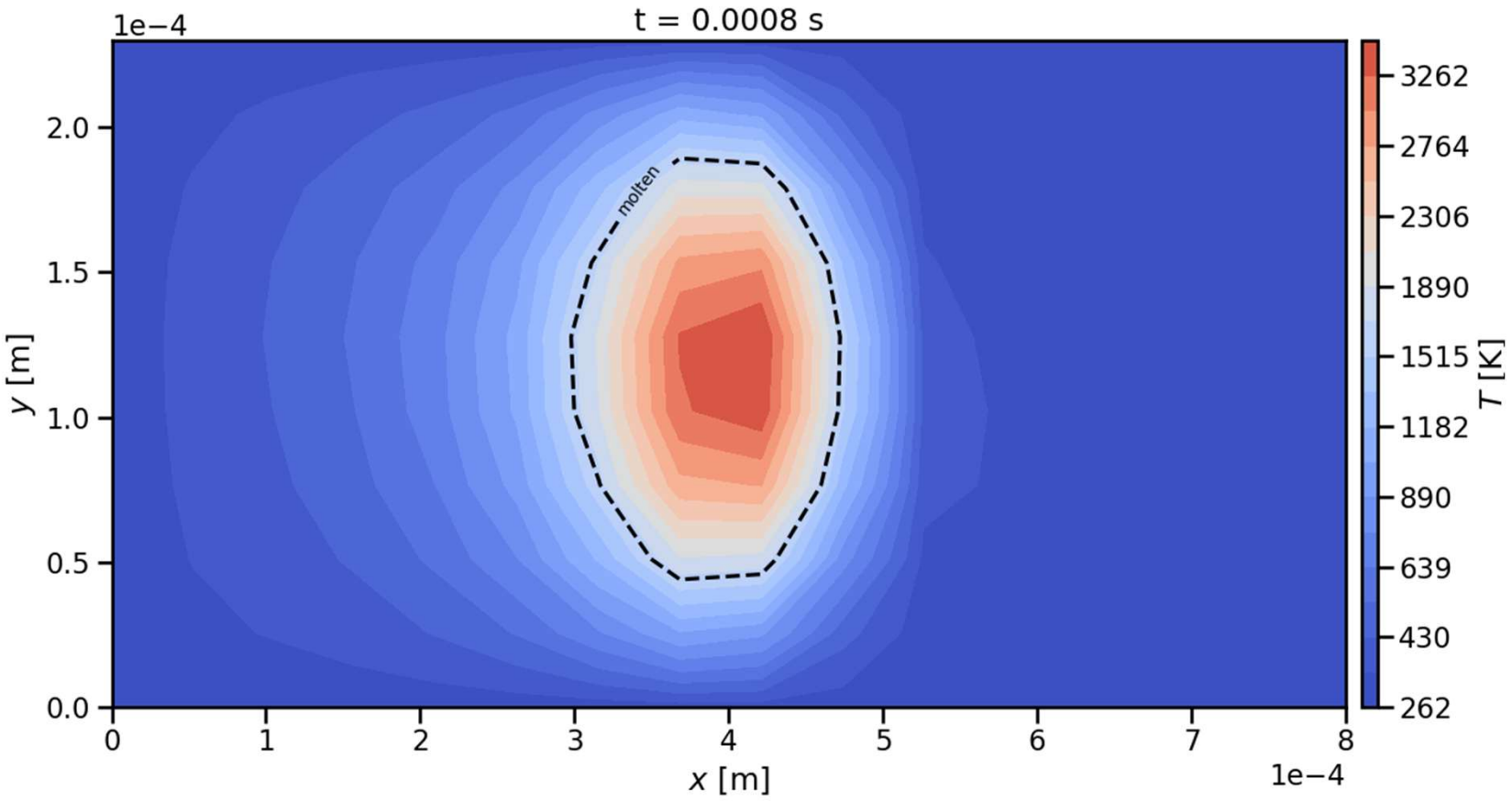
Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.



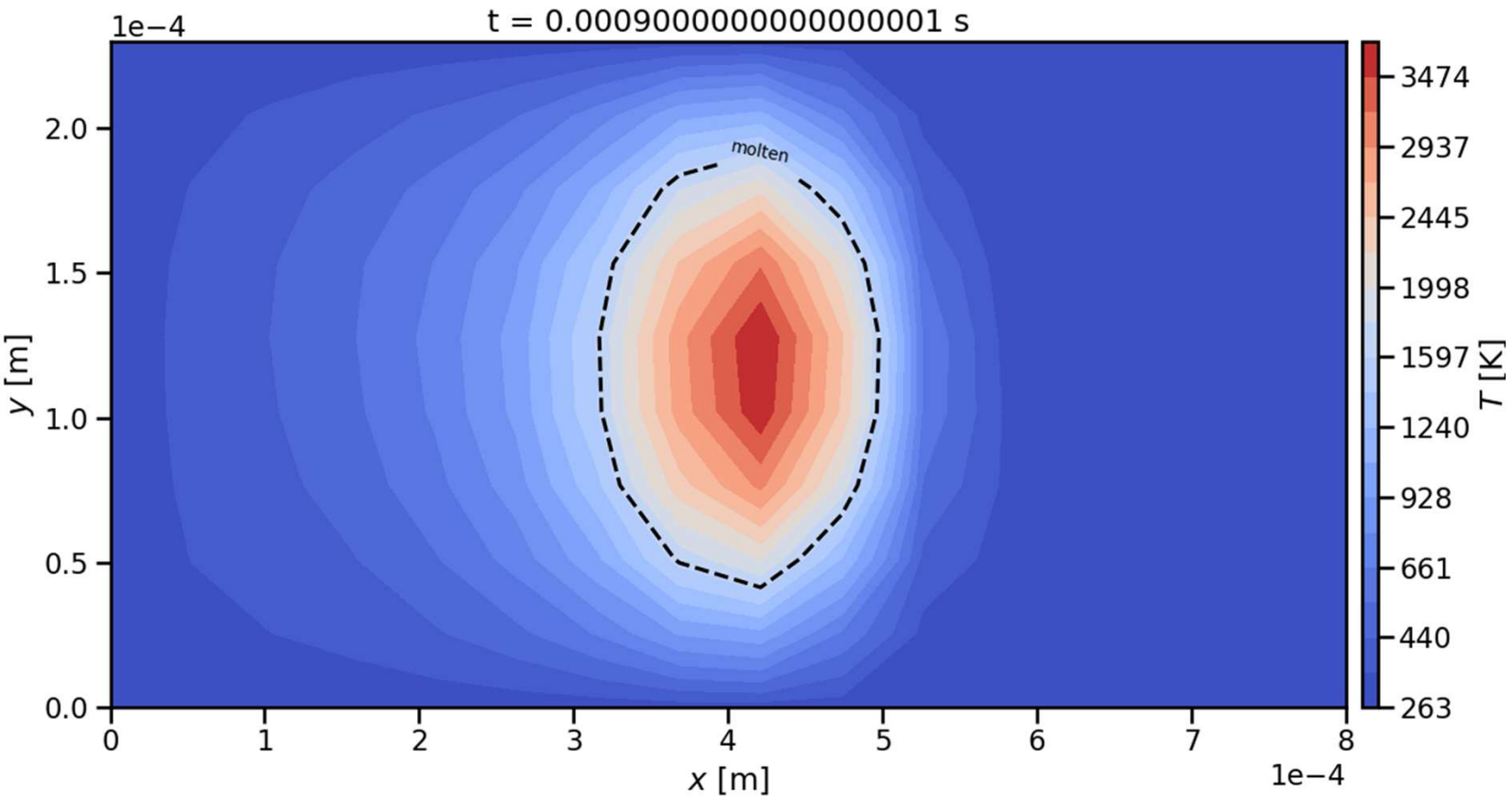
Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.



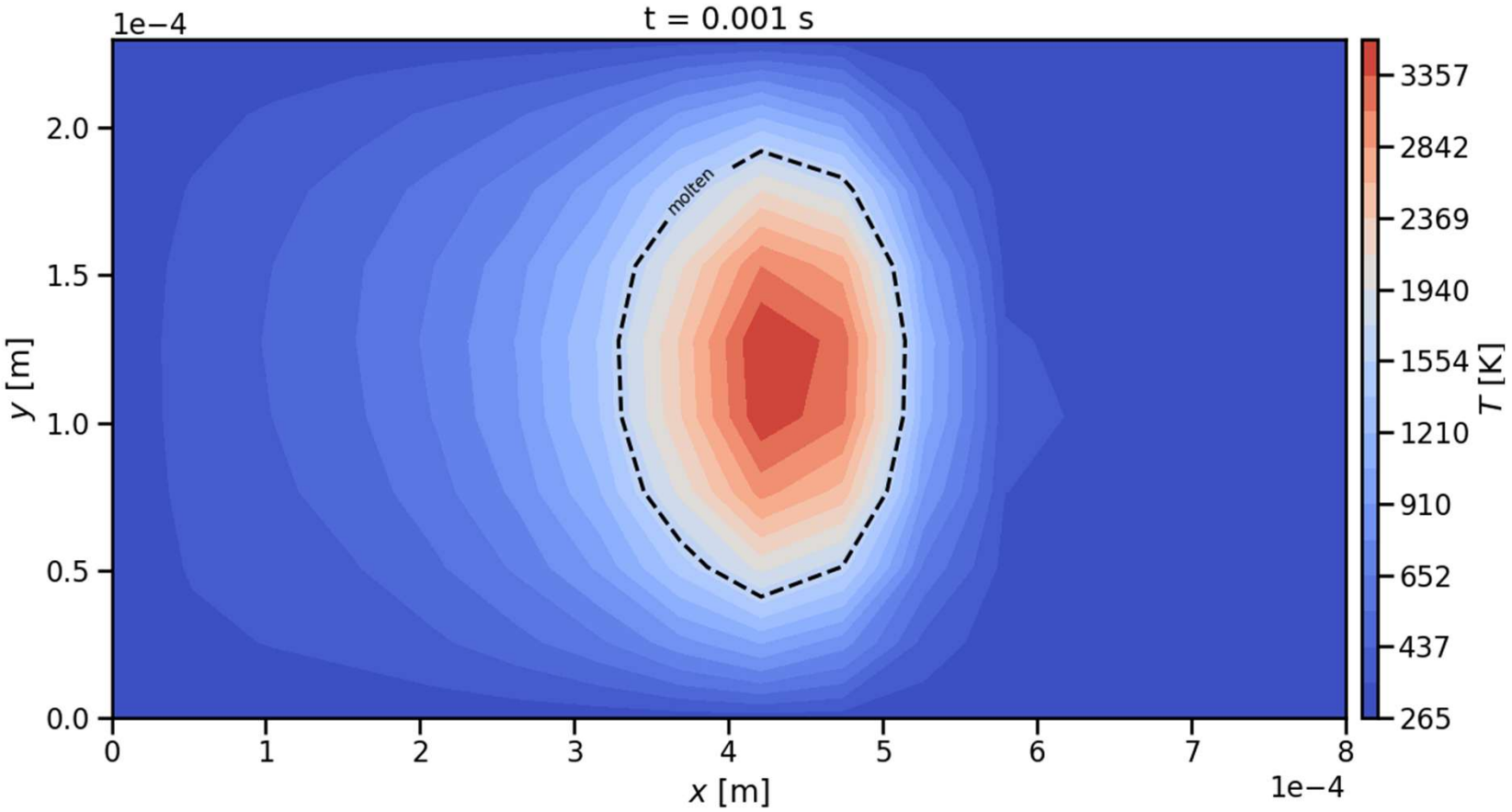
Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.



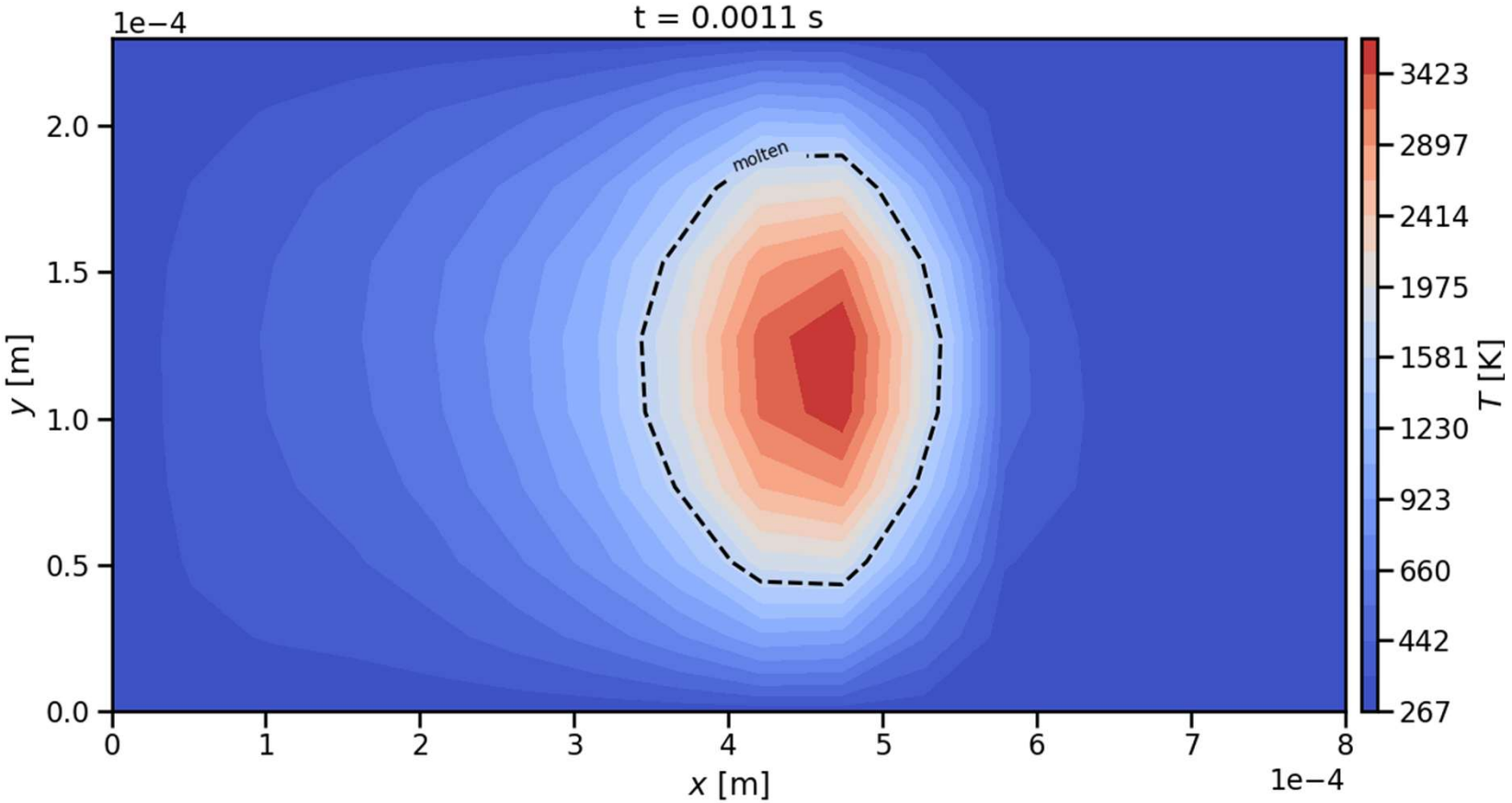
Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.



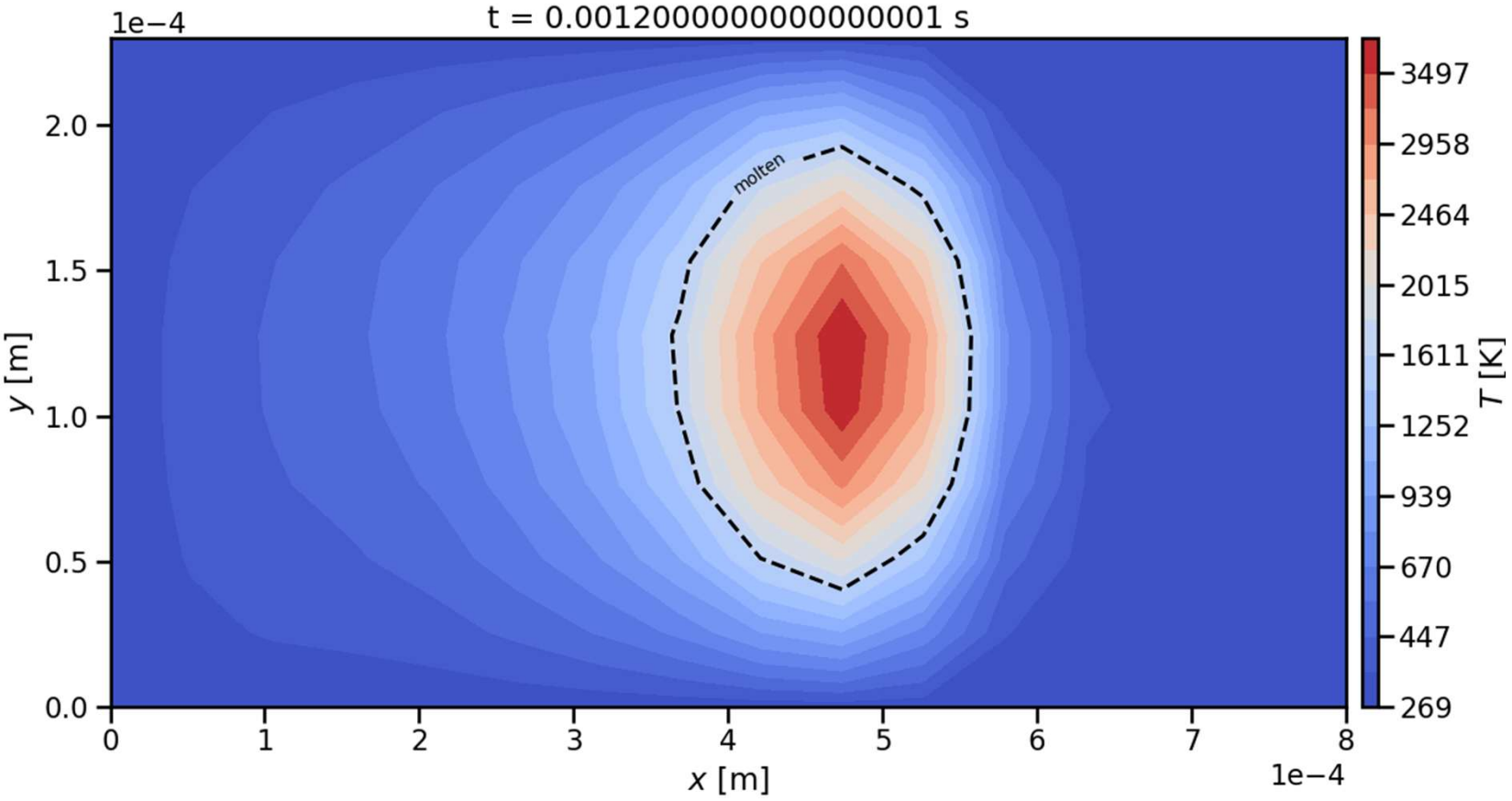
Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.



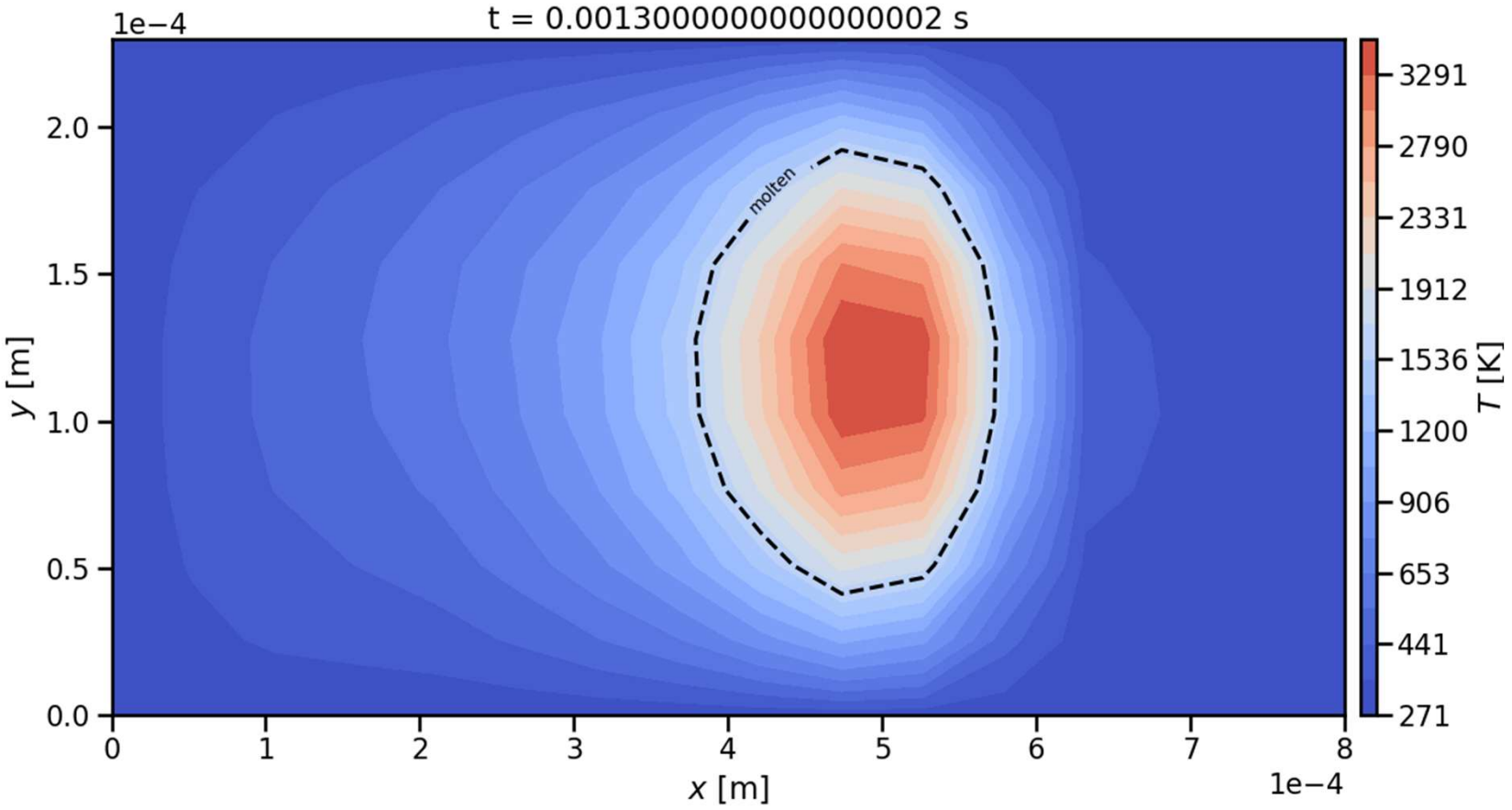
Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.



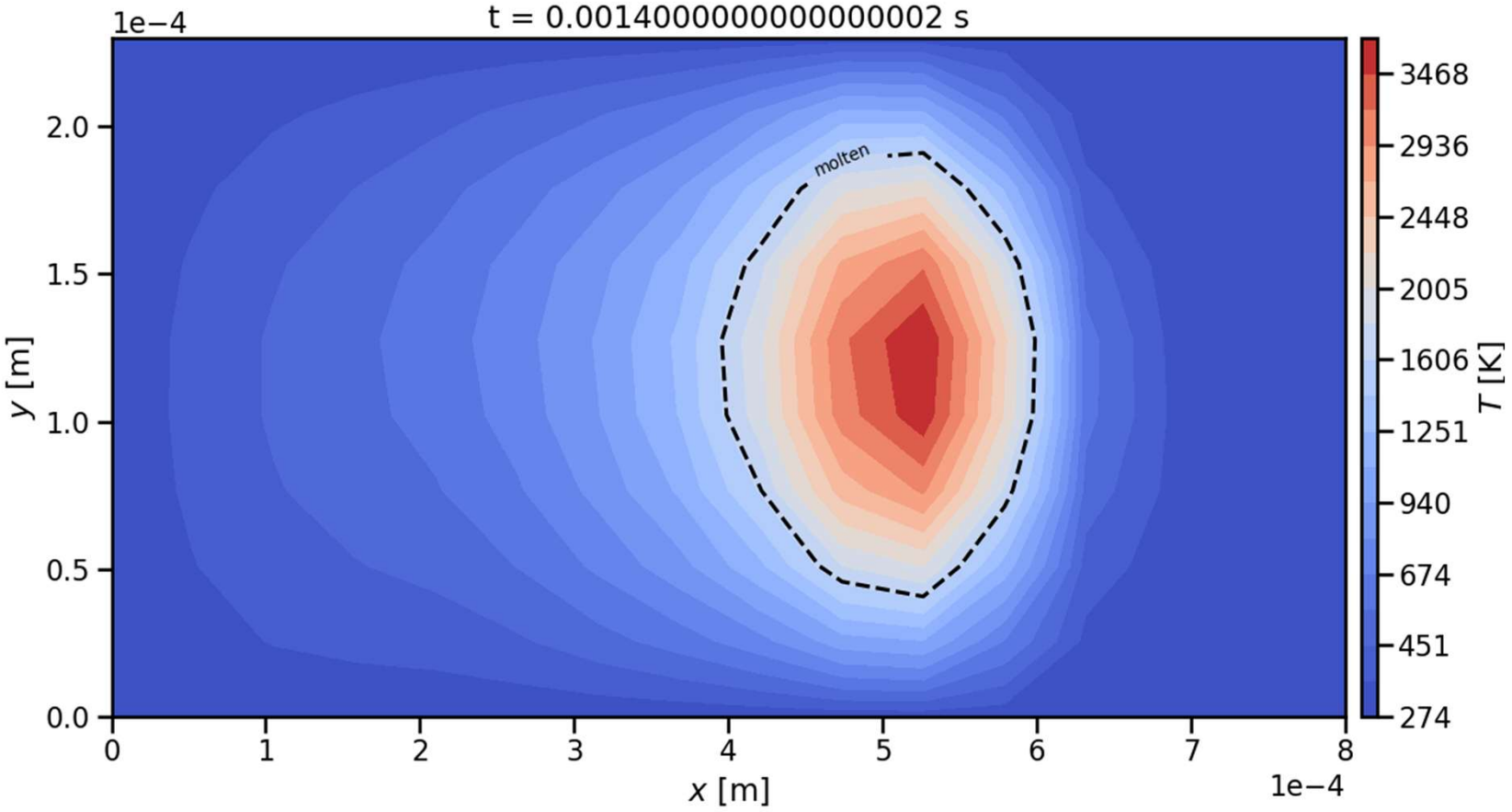
Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.



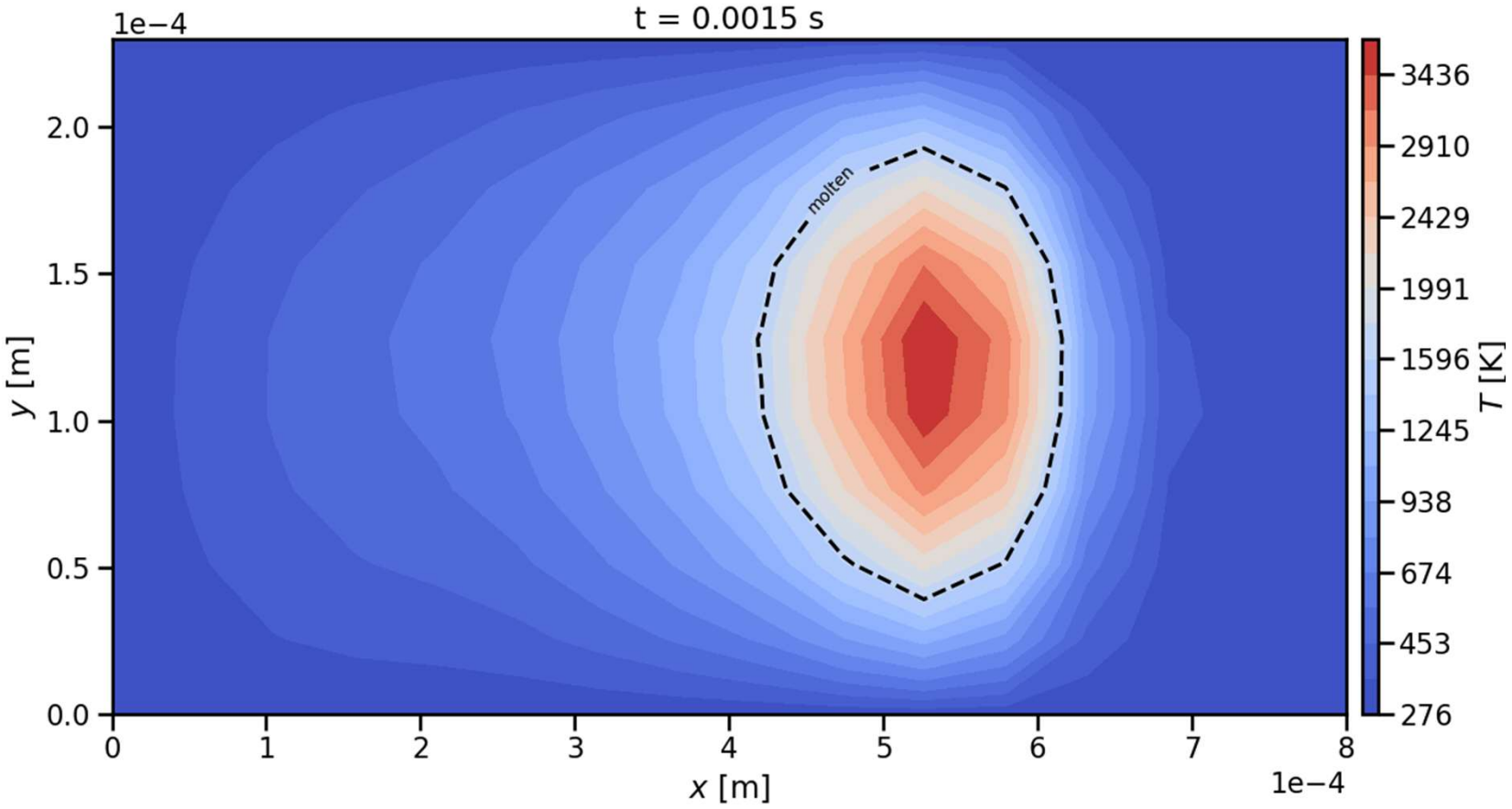
Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.



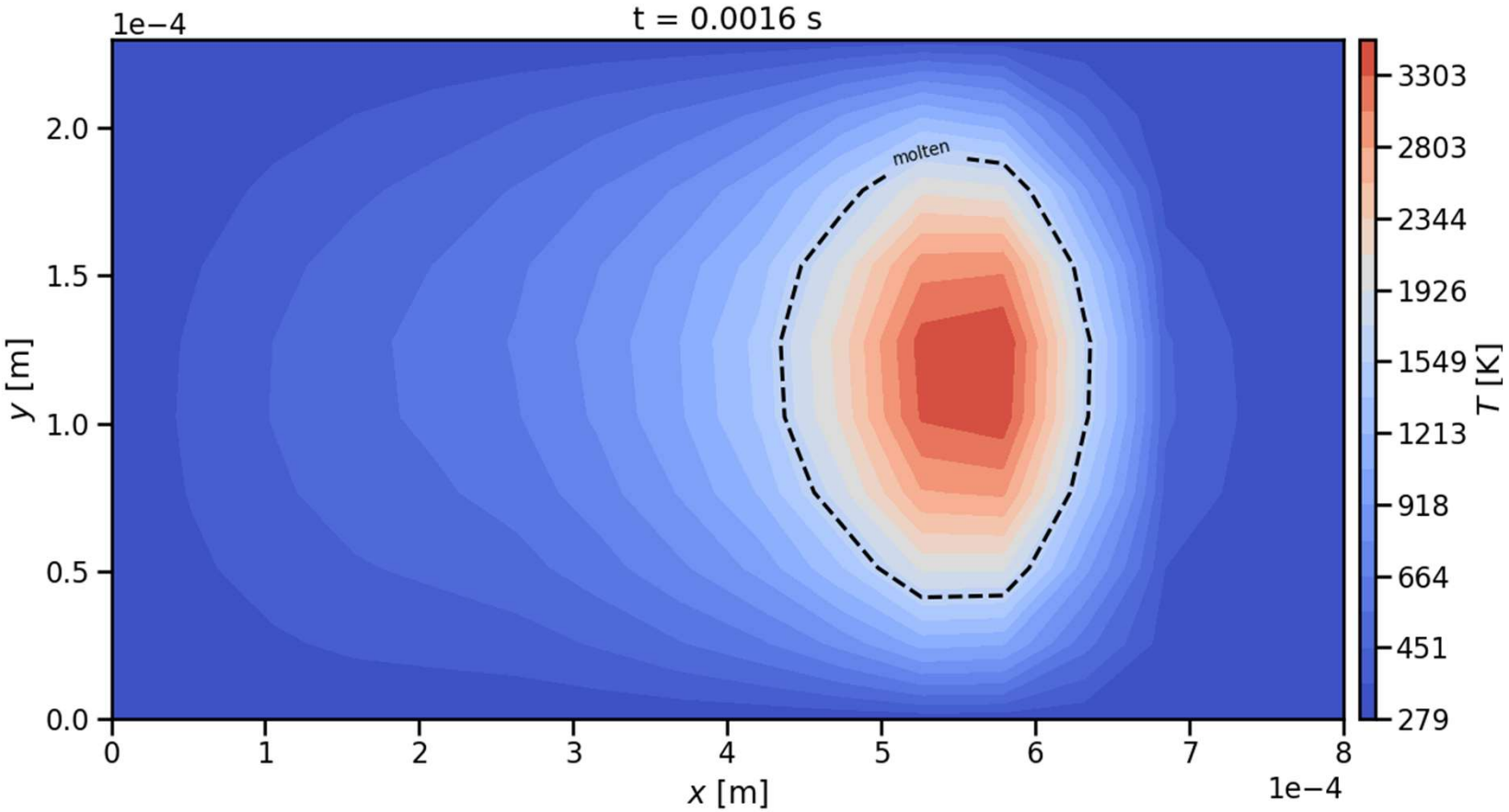
Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.



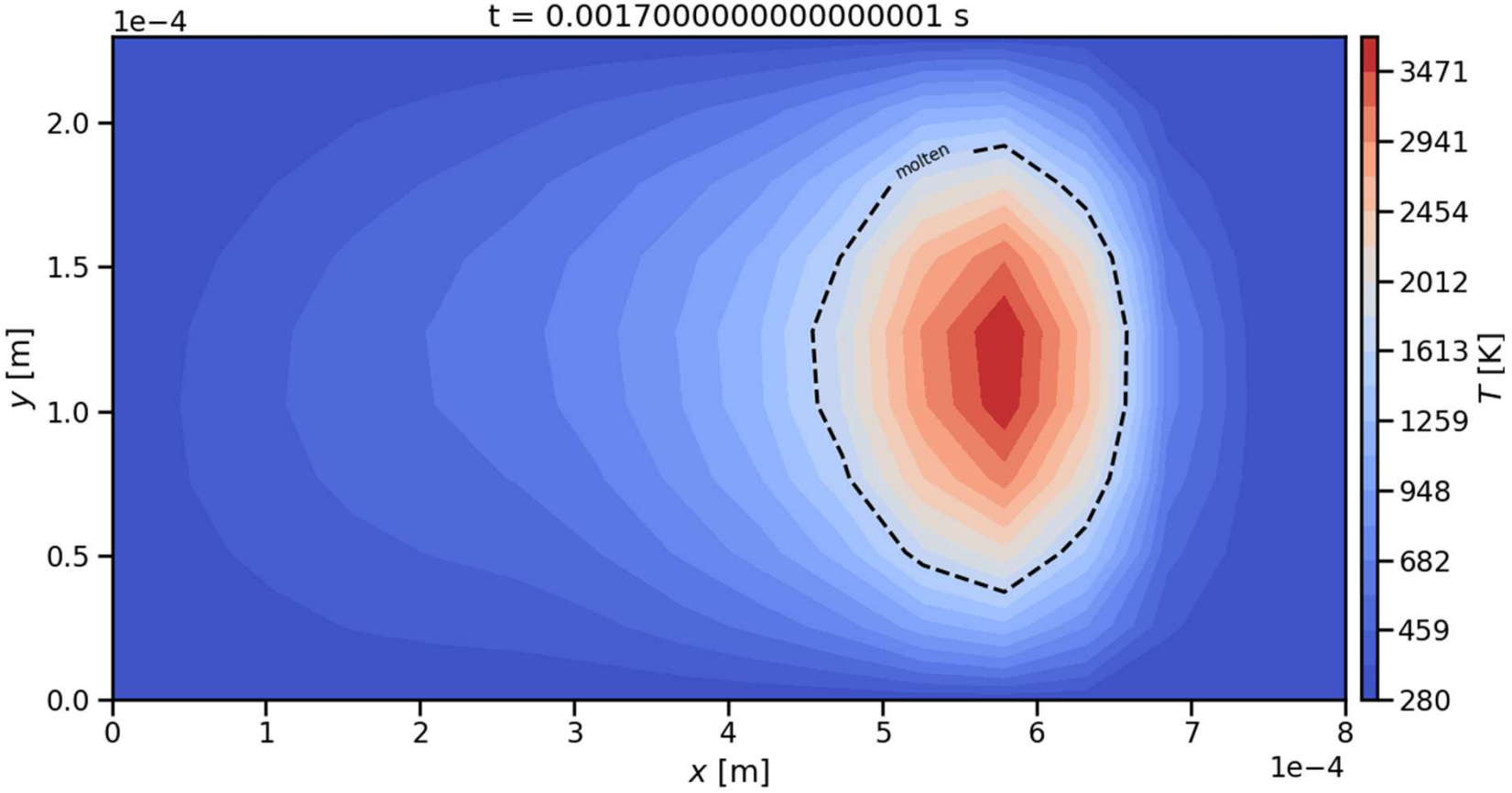
Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.



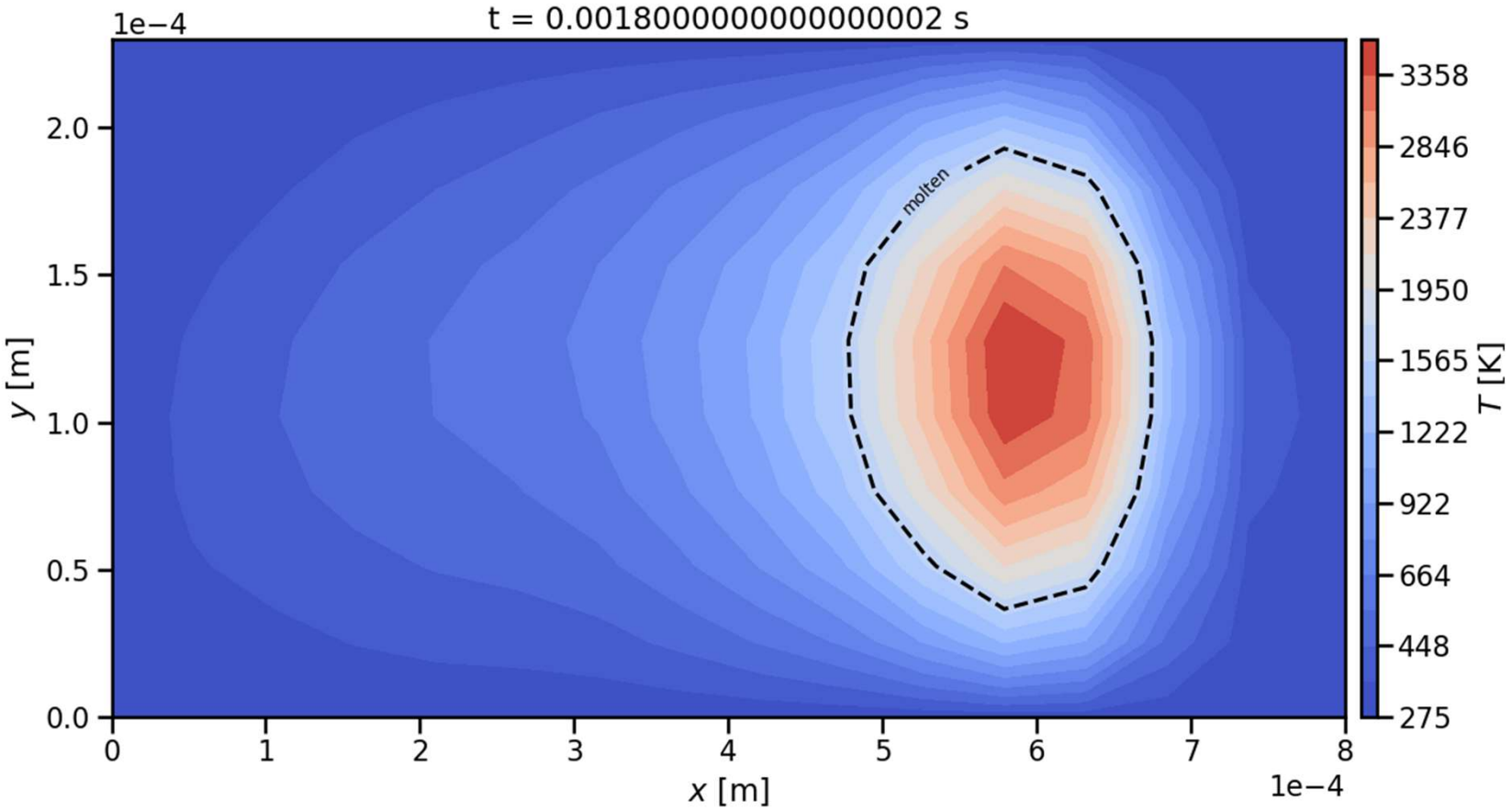
Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.



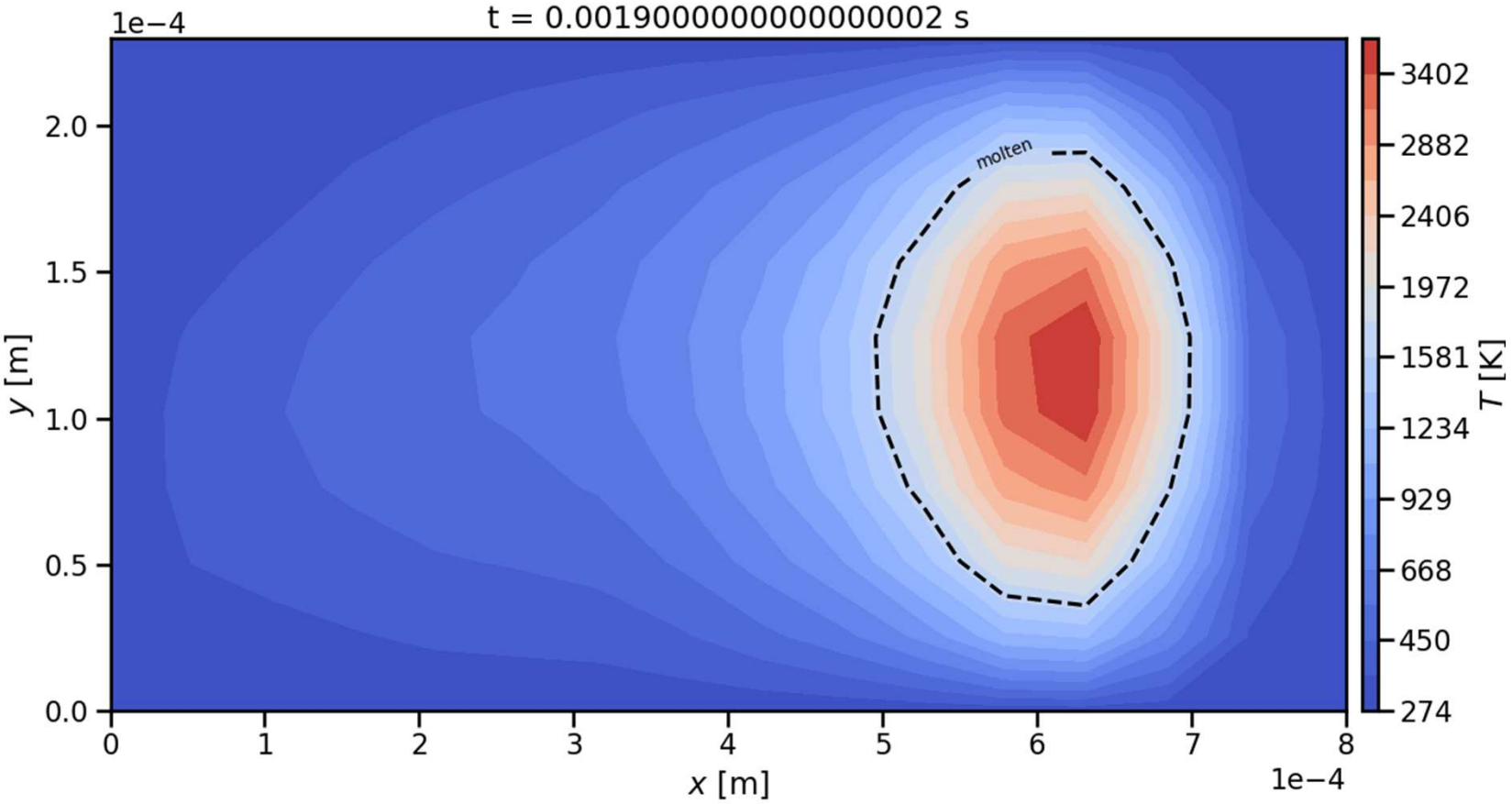
Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.



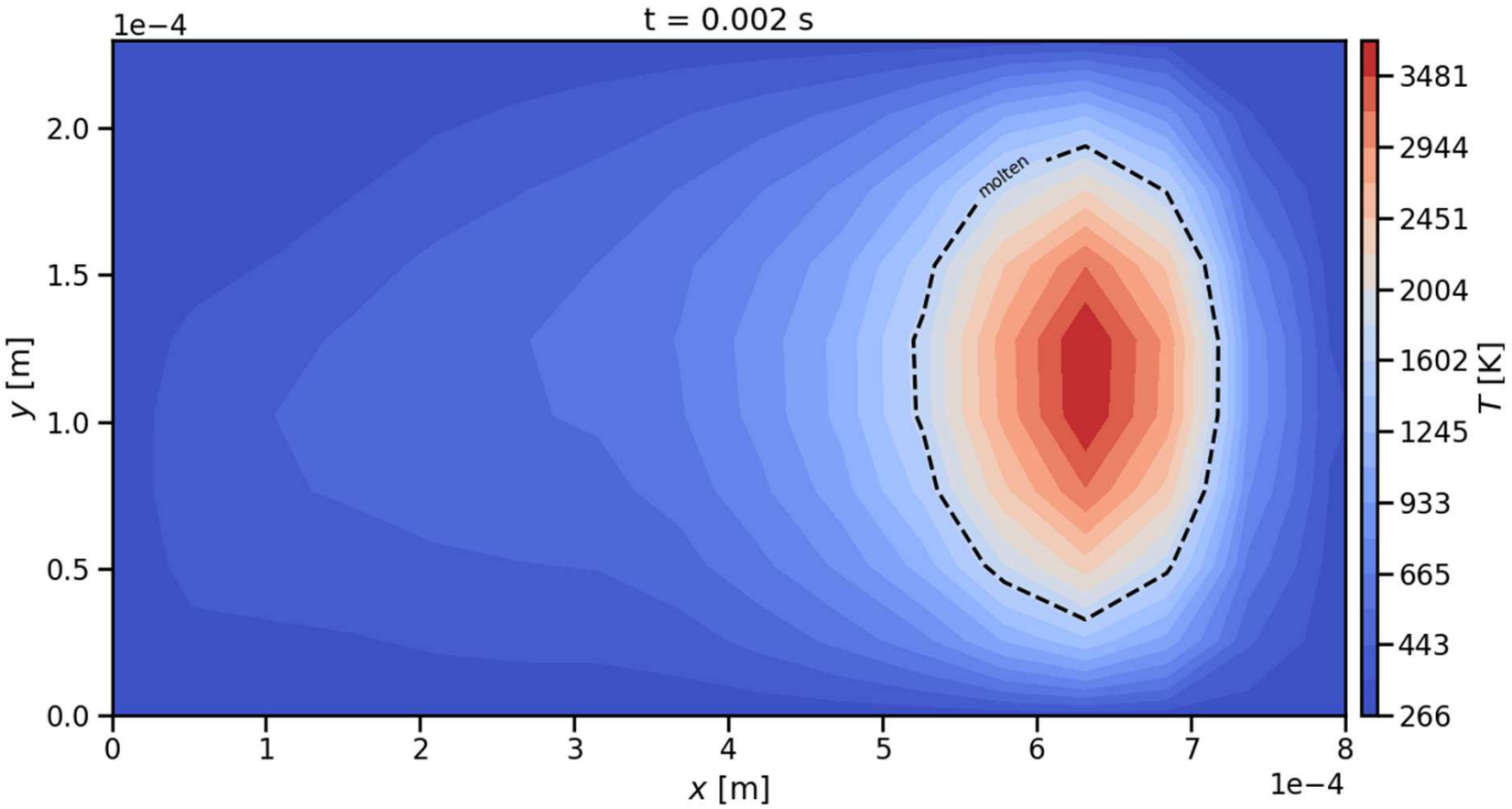
Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.



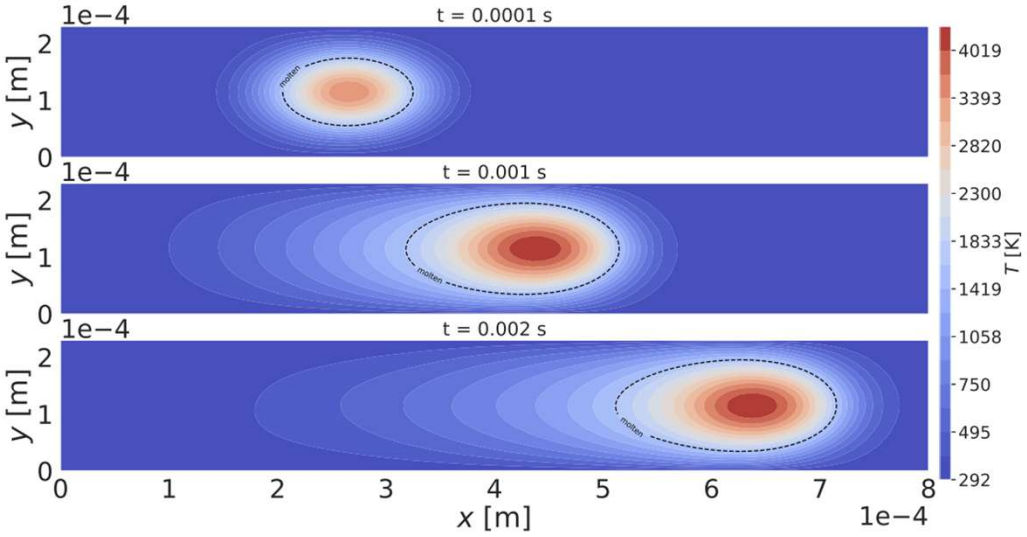
Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.

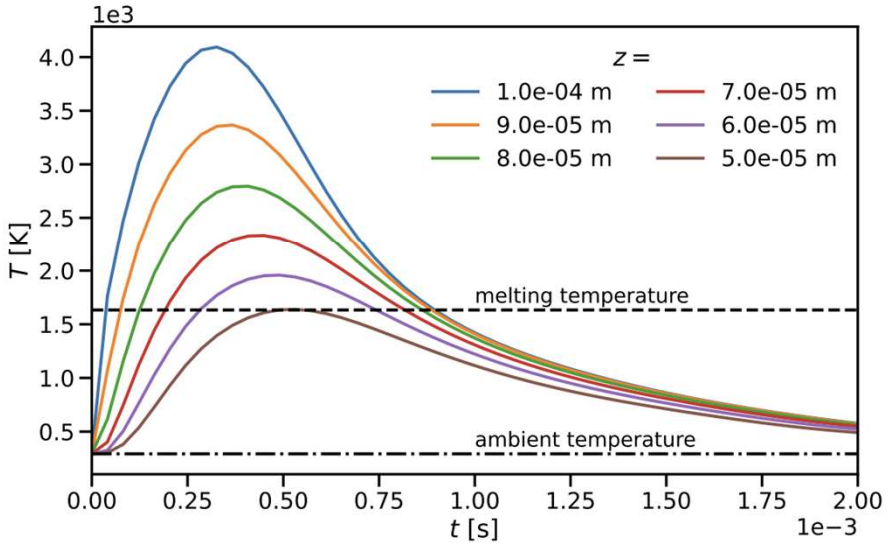


Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.



Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.





Uhrich, Benjamin, et al. "Physics-informed deep learning to quantify anomalies for real-time fault mitigation in 3D printing: B. Uhrich et al." *Applied Intelligence* 54.6 (2024): 4736-4755.



Exotische Differentialgleichungen

Zweipunkt-Randwertproblem



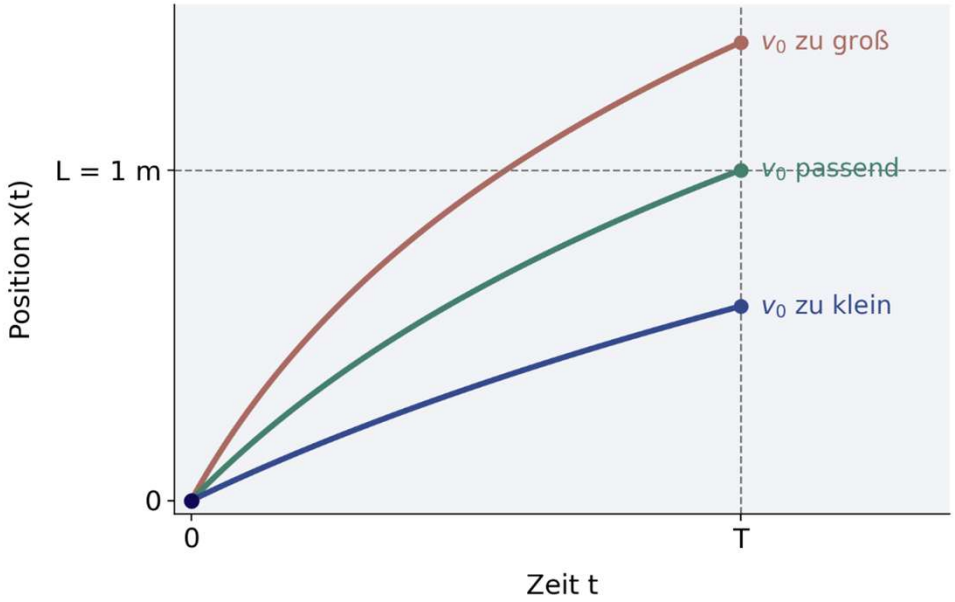
Du bewegst einen Ball mit der Masse m. Der Ball soll 1 m entfernt ankommen.
Was ist seine Anfangsgeschwindigkeit ?

$$\begin{cases} m\ddot{x}(t) = -\lambda \dot{x}(t)^2, \\ x(0) = 0, \\ \dot{x}(0) = v_0. \end{cases}$$

Numerische Methode?

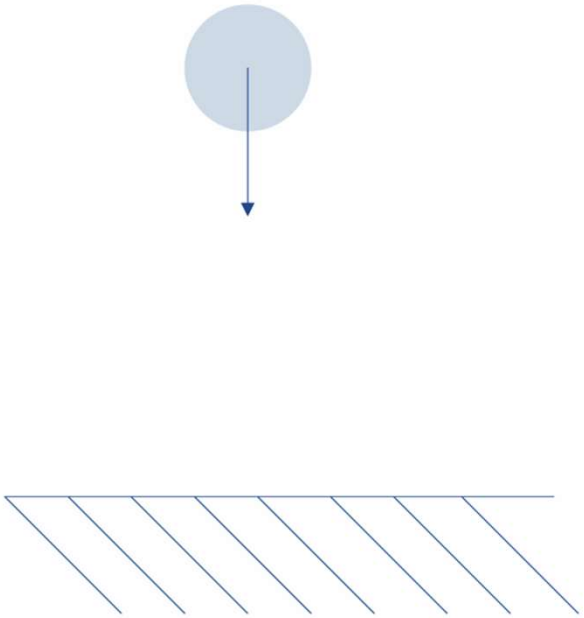
$$\begin{cases} X(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}, \\ \dot{X}(t) = g(t, X), \\ x_1(0) = x_1^0, \\ x_2(T) = x_2^T \end{cases}$$

Schießmethode



Zielabweichung $F(v_0) = x(T, v_0) - L$

Hybrid Systeme



$$x(t) > 0$$

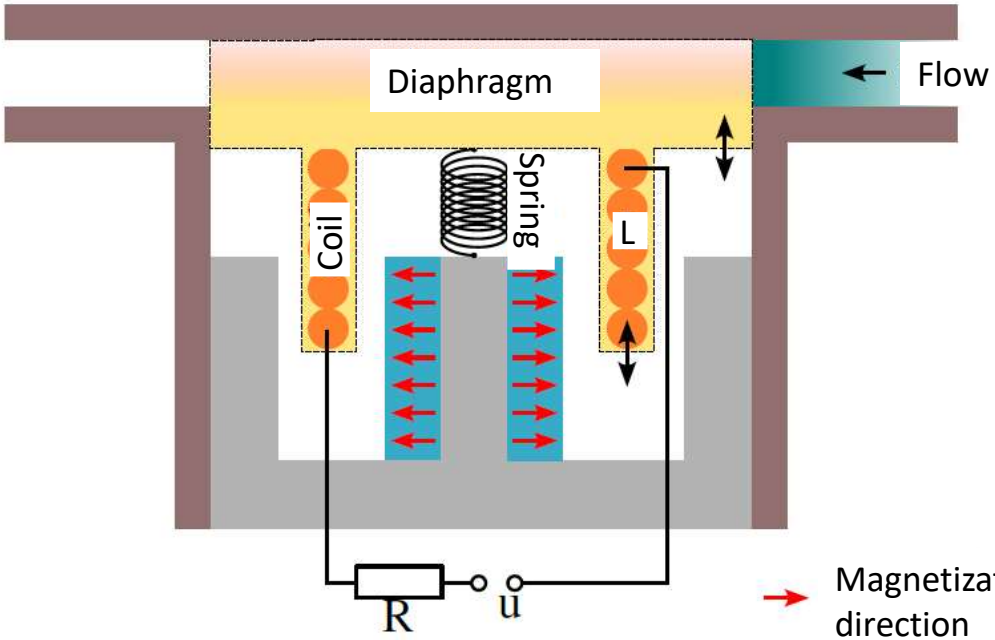
$$\begin{cases} \ddot{x}(t) = -g, \\ x(0) = 0, \\ \dot{x}(0) = v_0. \end{cases}$$

$$x(t) = 0$$

$$\begin{cases} \dot{x}(t^+) = -\mu v_t, \\ x(t) = 0, \\ \dot{x}(t) = v_t. \end{cases}$$

- Numerisches Schema sollte Übergänge prüfen

Gekoppelte Systeme



$$M_{CD}\frac{\mathrm{d}^2 x_D(t)}{\mathrm{d}t^2} = f_{\mathrm{L}}(t) - R_{SD}\frac{\mathrm{d}x_D(t)}{\mathrm{d}t} - K_{SDA}x_{\mathrm{D}}(t) \quad (1)$$

$$u_S(t) = u_{\mathrm{L}}(t) + L_C\frac{\mathrm{d}i_C(t)}{\mathrm{d}t} + R_C i_C(t) \quad (2)$$

$$f_L(t) = i_{\mathrm{C}}(t)\cdot\ell\cdot B \quad (3)$$

$$u_L(t) = \frac{\mathrm{d}x_D(t)}{\mathrm{d}t}\cdot B\cdot\ell \quad (4)$$

Uhrich, Benjamin, et al. "Using differential equation inspired machine learning for valve faults prediction." *2023 IEEE 21st International Conference on Industrial Informatics (INDIN)*. IEEE, 2023.

Zusammenfassung

