

Fussbodenheizung

Martin Kleindienst, Christoph Planitzer

25. Juni 2009

Inhaltsverzeichnis

1	Aufgabenstellung	1
2	Modellierung	2
2.1	Raum	2
2.2	Wärmeleitung	3
2.3	Rohre	3
3	Diskretisierung	4
3.1	erster Ansatz: explizites Euler-Verfahren	4
3.2	Implizites Euler-Verfahren	8
4	Der Algorithmus	10
4.1	Anfangskonfiguration	10
4.2	Restriktionen	10
4.3	Qualität der Wärmeverteilung	10
4.4	Iteration	11
5	Bunte Bilder	11
5.1	Die Anfangskonfiguration	11
5.2	Der Iterationsprozess	12
5.3	Entwicklungsschritte	18
6	Conclusio	19
6.1	Ergebnis	19
6.2	Ausbaumöglichkeiten	19

1 Aufgabenstellung

Unsere Aufgabe war es, ein Modell einer Fußbodenheizung aufzustellen. Selbstgestelltes Ziel des Modells ist dabei, eine möglichst optimale Verlegung der Rohre zum Erreichen einer angenehmen, gleichmäßigen Temperatur in einem kleinen, einfachen Raum wie etwa einem Bade- oder Wohnzimmer nach einer bestimmten oder nach quasi unendlicher Zeit.

2 Modellierung

2.1 Raum

Nach Abwägung des Aufwands gegen die Vor- und Nachteile haben wir uns für folgendes Modell entschieden:

Die Wärmeleitung im Boden wird durch eine zweidimensionale numerische Lösung der Wärmeleitgleichung genähert. Da Isoliermaterial unter den Rohren und die Luft im Raum über den Rohren das Modell (unnötig) verkomplizieren und der Wärmeverlust in diese dritte Dimension über der gesamten Bodenfläche näherungsweise proportional zur Bodentemperatur sein dürfte, ist im Modell ein konstanter Abgabefaktor eingebaut, der aus empirischen Daten ermittelt werden kann.

Von Interesse ist somit hauptsächlich die Diffusion der Temperatur im Boden selbst. Wir gehen von einem homogenen Baumaterial mit einem temperaturunabhängigen Wärmeleitkoeffizienten aus. Das dürfte bei den geringen Temperaturschwankungen in unserem Modellgebiet und einem normalen Zimmerboden auch annähernd gegeben sein.

Die physikalischen Grenzen des Modells sind also die folgenden:

Temperatur: Sie bewegt sich stets zwischen einem T_{min} (Temperatur wenn der Raum ausgekühlt ist) und T_{max} (höchstwahrscheinlich die Temperatur der Heizflüssigkeit)

Dass die Temperatur nie aus diesen Grenzen ausbrechen kann oder darf, geht aus der Natur des Problems hervor: Wärme kann sich nicht kumulieren sondern bloß verschmieren.

Raum: Wir gehen von einem beschränkten Raum von $a*b$ Quadratmetern aus.

Der Raum hat Wände mit gleicher Wärmeleitfähigkeit wie der Boden und konstanter Temperatur als Grenzen.

Zeit: Beginnend bei einer Zeit $t_0 = 0$ läuft die Zeitachse ausschließlich vorwärts bis zu einem t_{fin} oder $t = \infty$.

Für das gesamte Projekt wurden (auch in der Programmierung) ausschließlich SI-Einheiten verwendet. Lediglich die Temperatur lässt auch eine Eingabe in Celsius zu, da es sich hier um die selbe Skalierung handelt und das Programm nur mit Differenzen arbeitet.

2.2 Wärmeleitung

$$\nabla T(t) = \kappa \Delta T(\vec{x}) \quad (1)$$

Dies ist die allgemeine Wärmeleitgleichung für nicht stationäre Probleme mit dem (konstanten) Wärmeleitkoeffizienten κ . Es handelt sich dabei um eine typische parabolische Differentialgleichung, die Randbedingungen und eine Anfangstemperaturverteilung zum Zeitpunkt T_0 benötigt.

Die stationäre Version davon ist die Laplace-Gleichung:

$$\Delta T(\vec{x}) = 0 \quad (2)$$

Diese Gleichung ist elliptisch und kommt somit ohne Anfangswerte aus und benötigt lediglich Randbedingungen.

Wir haben länger diskutiert welche der beiden Gleichungen wir diskretisieren wollen, da einerseits die stationäre Lösung einfacher zu berechnen wäre, andererseits allerdings niemand gerne unendlich lange wartet bis sein Bad geheizt ist.

Da die Wärmeverteilung nach beispielsweise 10 Minuten sich jedoch noch von der stationären Lösung unterscheidet, ist unsere Entscheidung schlussendlich auf die allgemeine Wärmeleitgleichung (1) gefallen. Wir benötigen für unsere Zwecke die 2D-Version (mit leichten Modifikationen).

2.3 Rohre

Wir wollen die optimale Verlegung der Rohre im Boden erreichen, doch eine analytische Lösung steht uns hierfür nicht zur Verfügung.

Also arbeiten wir mit einem “Trial & Error“ - Verfahren, was in solchen Fällen durchaus üblich ist. Wir berechnen per Zufall aus einer Startkonfiguration eine neue, aus dieser wieder eine neue und so weiter. Bei jeder errechneten Konfiguration überprüfen wir, ob sie uns einer besseren Lösung näher bringt und so tasten wir uns langsam an eine (fast) optimale Lösung heran.

3 Diskretisierung

Für die Lösung der Wärmeleitgleichung wird in der Praxis üblicherweise die Finite-Elemente-Methode eingesetzt. Da wir es hier aber mit einem homogenen, rechteckigen Modellgebiet zu tun haben, reicht für unsere Zwecke die (viel einfachere) Finite-Differenzen-Methode vollständig aus. Hier haben wir auch zwei verschiedene Ansätze ausprobiert:

3.1 erster Ansatz: explizites Euler-Verfahren

Der einfachst mögliche und damit erste Ansatz ist die Diskretisierung mittels explizitem Euler-Verfahren. Die erste Ableitung nach der Zeit wird dabei durch den Differenzenquotienten und die zweite Ableitung im Raum durch den zentralen doppelten Differenzenquotienten ersetzt. In der expliziten Form zieht man dafür die Werte aus dem vorangehenden Zeitschritt heran.

Die ausführliche Schreibweise der 2D-Wärmeleitgleichung (1) lautet:

$$\frac{d}{dt}T(t) = \frac{d^2}{dx^2}T(x, y) + \frac{d^2}{dy^2}T(x, y) \quad (3)$$

Explizit diskretisiert sieht diese Formel dann so aus:

$$\begin{aligned} \frac{T_{n+1} - T_n}{\Delta t} &= \kappa * \frac{T_n(x + \Delta x, y) - 2 * T_n(x, y) + T_n(x - \Delta x, y)}{(\Delta x)^2} + \\ &\quad \kappa * \frac{T_n(x, y + \Delta y) - 2 * T_n(x, y) + T_n(x, y - \Delta y)}{(\Delta y)^2} \end{aligned} \quad (4)$$

Man hat sich das ganze dann so vorzustellen: Das Modellgebiet wird räumlich in ein diskretes Gitter 1 unterteilt, dem eine Anfangstemperatur zugewiesen wird. Die Zeit wird in Form einzelner Zeitschritte diskretisiert, die jeder einem Iterationsschritt im Euler-Verfahren entsprechen.

Da in unserem Modell aber nicht nur Wärmeleitung sondern auch Konvektion auftritt, muss man Gleichung (4) noch um einen Term erweitern, der die Konvektionsgeschwindigkeit einbringt:

$$(4) + v_x * \frac{T(x \pm \Delta x, y)}{\Delta x} + v_y * \frac{T(x, y \pm \Delta y)}{\Delta y} \quad (5)$$

Man bemerke: Da man hier nicht die zweite sondern die erste Ableitung im Raum diskretisiert, ist die Formel nicht symmetrisch. Es kommt daher

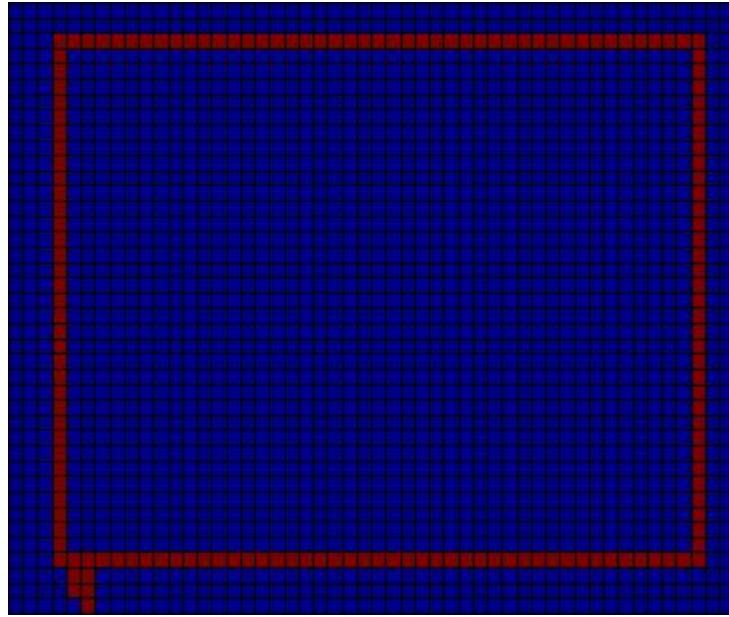


Abbildung 1: Diskretisierter Raum mit Beispielrohren (rot)

sozusagen darauf an, ob man “aus der richtigen Richtung“ diskretisiert, da es sonst insbesondere beim expliziten Euler-Verfahren schnell zu unerwünschten Oszillationen kommen kann. Das \pm deutet an, dass hier eigentlich vier Terme hingehören, von denen in unserem Modell mindestens drei jedoch null sind (Das Wasser kann nur aus einer Richtung nachfließen).

Diese Form führt auf eine einfache Funktion $T_{n+1} = F(T_n)$ bzw.

$$\overrightarrow{T_{n+1}} = A * \overrightarrow{T_n} \quad (6)$$

Der Vektor $\overrightarrow{T_n}$ ist dabei ein Vektor aller einzelnen Pixel in die der Raum unterteilt worden ist. Die A ist für das einfache Modellgebiet schnell aufgestellt. Es handelt sich um eine dünnbesetzte Diagonalmatrix. Das leuchtet unmittelbar ein, wenn man bedenkt, dass jedes Pixel nur von seinen (vier) Nachbarpixeln beeinflusst wird.

Wir haben für unser Modell $\Delta x = \Delta y$ gewählt.

$$A = \begin{pmatrix} d & r & \cdots & u & \cdots & 0 \\ l & d & r & \cdots & \ddots & \vdots \\ \vdots & l & d & r & \cdots & u \\ o & \cdots & l & d & r & \vdots \\ \vdots & \ddots & \cdots & \ddots & \ddots & \vdots \\ 0 & \cdots & o & \cdots & l & d \end{pmatrix} \quad (7)$$

mit

$$d = 1 - (r + l + o + u)$$

$$r = \kappa * \frac{\Delta t}{(\Delta x)^2} + v_{n,r} * \frac{\Delta t}{\Delta x}$$

l, u, o analog mit dem entsprechenden $v_{n,i}$

r, l, u, o sind hier die Koeffizienten die das Pixel mit seinem rechten, linken, unteren und oberen Nachbarn verbinden. Sie sind alle gleich bis auf den Konvektionsterm der in jeder Zeile nur maximal einmal vorkommt. Im wesentlichen ist also

$$d = 1 - 4 * \kappa * \frac{\Delta t}{(\Delta x)^2} - \text{Konvektionsterm} \quad (8)$$

Diese Matrix-Vektor-Mulitplikation (??) berechnet man nun mehrmals um bis zum gewünschten Zeitpunkt hinzuerrechnen. Sofort stellt sich natürlich die Frage nach der Größe der Zeitschritte und hier kommen *Konsistenz* und *Stabilität* ins Spiel.

Konsistenz

Gegeben sei ein kontinuierliches Problem und die exakte Lösung $u(t)$ sowie die numerische Lösung $u_h(t)$ zu einer Schrittweite $h > 0$. Das Verfahren heißt konsistent, falls es eine Funktion $\sigma(h)$ mit $\lim_{h \rightarrow 0} \sigma(h) = 0$ gibt, sodass für den lokalen Fehler gilt:

$$\|u(t_h) - u_h(t_h)\| \leq \sigma(h) \quad \forall t_h = n * h \quad (9)$$

Es besitzt eine Konsistenzordnung $p \in \mathbb{N}$ falls $\sigma(h) = \mathcal{O}(h^p)$ Das bedeutet, dass man zu jedem Zeitpunkt eine Fehlerbeschränkung in Abhängigkeit der gewählten Schrittweite hat und dieser Fehler gegen Null geht wenn die Schrittweite gegen Null geht.

Für das explizite Euler-Verfahren ist die Konsistenzordnung schnell gezeigt. Für die Zeitableitung wird der einfache Differenzenquotient (4) verwendet:

$$f'(t) \approx \frac{f(t + \Delta t) - f(t)}{\Delta t} \quad (10)$$

Taylorentwicklungen:

$$f(t + \Delta t) = f(t) + f'(t) * \Delta t + \frac{f''(\zeta) * (\Delta t)^2}{2} \quad (11)$$

woraus folgt

$$f'_{\Delta t}(t) = \frac{f(t + \Delta t) - f(t)}{\Delta t} = f'(t) + \frac{f''(\zeta) * \Delta t}{2} \quad (12)$$

$$\|f'(t) - f'_{\Delta t}(t)\| = \frac{f''(\zeta) * \Delta t}{2} = \mathcal{O}((\Delta t)^1) \quad (13)$$

Diese Implementierung hat also die Konsistenzordnung $P = 1$!

Stabilität

Ein Verfahren heißt stabil, wenn es gegenüber kleinen Störungen der Daten unempfindlich ist. Insbesondere bedeutet dies, dass sich Rundungsfehler nicht zu stark auf die Berechnungen auswirken.

Wir überprüfen das explizite Euler-Verfahren auf A-Stabilität und wenden die folgende Testgleichung an:

$$y' = \lambda * y \quad (14)$$

mit

$$y(0) = 1 \quad (15)$$

$$T'(t) \approx \frac{T_{k+1} - T_k}{\Delta t} = \lambda * T_k \quad (16)$$

$$\begin{aligned} T_{k+1} &= T_k + \Delta t * \lambda * T_k = (1 + \Delta t * \lambda) * T_k \\ &= (1 + \Delta t * \lambda)^k * T_0 = (1 + \Delta t * \lambda)^k \end{aligned} \quad (17)$$

Das Verfahren ist also für $\Delta t * \lambda$ innerhalb eines Kreises mit Radius 1 um den Mittelpunkt -1 in der komplexen Zahlenebene stabil, folglich also nicht A-stabil (Das wäre es, wenn es für ganz \mathbb{C}^- stabil wäre).

Die Schlussfolgerung daraus: Man kann für das explizite Euler-Verfahren keine beliebig große (Zeit-)Schrittweite wählen. In unserem Fall darf d aus (??) auf keinen Fall negativ, im Idealfall nicht kleiner als $\frac{1}{2}$ werden (Dann ist die Matrix A (7) noch diagonaldominant). Im weiteren Sinne hat das die Bedeutung, dass ein Eintrag sich in einem Zeitschritt nicht um mehr ändern darf als er selber ist, bzw. idealerweise nicht um mehr als die Hälfte. Für $d \in (0; 0, 5]$ können leichte Oszillationen auftreten, die aber möglicherweise das Ergebnis für $t \rightarrow \infty$ nicht beeinträchtigen, für $d < 0$ allerdings oszilliert das System über alle Grenzen.

3.2 Implizites Euler-Verfahren

Die Instabilität bei großen Zeitschritten ist natürlich mehr als störend, da das verwendete Verfahren zur Rohrverlegung eine große Anzahl von Simulationen braucht und diese deshalb keinen allzu großen Aufwand haben sollten. Wir suchen deshalb nach einem A-stabilen Verfahren.

Man vergleiche die untenstehende Gleichung (19) für das implizite Euler-Verfahren mit der für das explizite Euler-Verfahren (4). Im Gegensatz zu letzterer zieht dieses Verfahren nicht die letzte bekannte Konfiguration heran sondern es ist nach dem Prinzip

$$T'(t) \approx \frac{T_{k+1} - T_k}{\Delta t} = \kappa * f(T_{k+1}) \quad (18)$$

in jedem Schritt ein lineares Gleichungssystem zu lösen.

Ziemlich analog zu (4) ergibt sich mit $\Delta x = \Delta y$ und inklusive Konvektionsterm:

$$\begin{aligned} \frac{T_{n+1} - T_n}{\Delta t} &= \frac{\kappa}{(\Delta x)^2} * (T_{n+1}(x + \Delta x, y) + T_{n+1}(x, y + \Delta x) - 4 * T_{n+1}(x, y) + \\ &T_{n+1}(x - \Delta x, y) + T_{n+1}(x, y - \Delta x)) + \frac{v_{x,y}}{\Delta x} * (T_{n+1}(x \pm \Delta x, y \pm \Delta y)) \end{aligned} \quad (19)$$

Statt (6) schreiben wir jetzt also das Gleichungssystem

$$A * \overrightarrow{T_{n+1}} = \overrightarrow{T_n} \quad (20)$$

Dieses Verfahren hat ebenfalls Konsistenzordnung $P = 1$. (siehe (13))

Hier führt die Stabilitätsberechnung allerdings auf

$$T_{k+1} = T_k - \Delta t * \lambda * T_{k+1} \quad (21)$$

$$T_k = (1 + \Delta t * \lambda) * T_{k+1}$$

$$T_k = (1 + \Delta t * \lambda)^{-k} * T_0 = (1 + \Delta t * \lambda)^{-k} \quad (22)$$

Man erkennt deutlich, dass das Verfahren A-Stabil ist, da die änderung der numerische Lösung unabhängig von der Wahl der Zeitschrittweite gegen Null konvergiert.

Das bringt uns jetzt den Vorteil, beliebig große Zeitschritte wählen zu können. Das beeinträchtigt zwar die Genauigkeit der Simulation, Versuche haben aber gezeigt, dass mit drei großen Zeitschritten die selbe Rohrkonfiguration herauskommt wie mit fünfzig - die Genauigkeit ist (innerhalb gewisser Grenzen) also relativ irrelevant für das Ergebnis.

Die Matrix A für das Gleichungssystem hat hier die gleiche Form wie bei (6), jedoch andere Einträge für die Diagonalen (vergleiche: (7), man beachte das negative Vorzeichen):

$$\begin{aligned} d &= 1 - (r + l + o + u) \quad (23) \\ r &= -\kappa * \frac{\Delta t}{(\Delta x)^2} - v_{n,r} * \frac{\Delta t}{\Delta x} \end{aligned}$$

l, u, o analog mit dem entsprechenden $v_{n,i}$

4 Der Algorithmus

Der Algorithmus der die verschiedenen Rohr-Konfigurationen aufbaut, versteht sich als Monte-Carlo-Methode. Es werden mit unterschiedlichen Wahrscheinlichkeiten verschiedene Änderungen an der vorhergehenden Konfiguration vorgenommen.

4.1 Anfangskonfiguration

Als Start-Konfiguration wählten wir zuerst einfach Rohre die direkt an der Wand entlang den ganzen Raum umschlossen, wie in Abbildung 1 zu sehen. Es stellte sich jedoch als vorteilhafter heraus, die Rohre einige Zentimeter weiter im Rauminneren anzusiedeln, da der Algorithmus dazu tendiert, die Rohre auf Dauer weiter nach innen zu verlegen. So ersparen wir uns einige Iterationsschritte.

4.2 Restriktionen

Im Prinzip besitzt der Algorithmus nur einen gewissen Satz an Änderungen denen er das Gitter unterziehen kann. Diese sind über eine Zufallszahl mit Wahrscheinlichkeiten versehen, die bestimmen wie häufig sie auftreten sollen. Zusätzlich haben wir eine Maximale Rohrlänge implementiert die von der Gittergröße abhängig ist. Ist die maximale Länge erreicht koennen nur mehr Operationen die das Rohrnetz verkürzen oder die Länge unverändert lassen ausgefuehrt werden, bis die Länge wieder unter ihren Maximalwert gefallen ist.

4.3 Qualität der Wärmeverteilung

Ist eine neue Konfiguration berechnet, muss bestimmt werden ob sie Vorteile gegenüber der alten hat. Dafür wird die Qualität der Wärmeverteilung im Raum berechnet. Um diese Qualität zu berechnen wird zuerst jeder Punkt im Simulationsraum nach Bedeutung gewichtet. Da die Temperatur in Wandnähe normalerweise weniger wichtig ist als die in der Mitte des Raums, implementierten wir diese Gewichtung mit 2 Sinusfunktionen (je eine halbe Periode zwischen 2 gegenüberliegenden Wänden). Danach wird für jeden Punkt das Quadrat der Differenz von einer gewünschten Idealtemperatur

berechnet und mit dem dazugehörigen Gewicht multipliziert. Kumuliert ergeben diese Werte ein Maß für die Qualität der Temperaturverteilung im Raum.

4.4 Iteration

Da dieser Wert nach Möglichkeit minimiert werden sollte, wird er bei jeder neu errechneten Konfiguration mit dem gespeicherten Wert der vorhergehenden Konfiguration verglichen und sollte er niedriger sein, wird sowohl der neue Wert als auch die neu errechnete Konfiguration gespeichert. Danach wird von dieser Konfiguration ausgegangen und so wird sich langsam einem sehr guten Ergebnis genähert. Da die Gefahr besteht in einem lokalen Minimum ‘hängenzubleiben’, werden auch Konfigurationen mit einer schlechteren Wärmeverteilung als der vorhergehenden in Betracht gezogen. Diese werden natürlich nicht immer genommen, sondern nur mit einer gewissen Wahrscheinlichkeit, die davon abhängt wieviel schlechter die gerade errechnete Konfiguration ist.

Mit diesem Verfahren kann zwar keine ideale Lösung berechnet werden (oder nur mit sehr viel Glück), aber man nähert sich auf jeden Fall einer sehr guten Lösung für das Problem. Das Prinzip dieses Algorithmus’ haben wir vom bekannten Metropolis-Algorithmus zur Simulation von Boltzmann-Verteilungen übernommen. Dieser ist zwar sehr einfach, das Prinzip konnten wir aber zur Simulation unseres komplexeren Problems adaptieren.

5 Bunte Bilder

Nun wollen wir natürlich auch ein Ergebnis präsentieren:

5.1 Die Anfangskonfiguration

Die Verlegung der Rohre war schon in der ersten Abbildung 1 zu sehen. Von diesem Punkt startet der Algorithmus.

Diese spezielle Anfangskonfiguration wurde gewählt, da der Algorithmus in Vorversuchen bereits dazu geneigt hat, sich an den Rand des Raumes anzulegen.

5.2 Der Iterationsprozess

Hier sehen Sie einen Beispielderationsprozess mit einer Gesamtdauer von etwa 10 min:

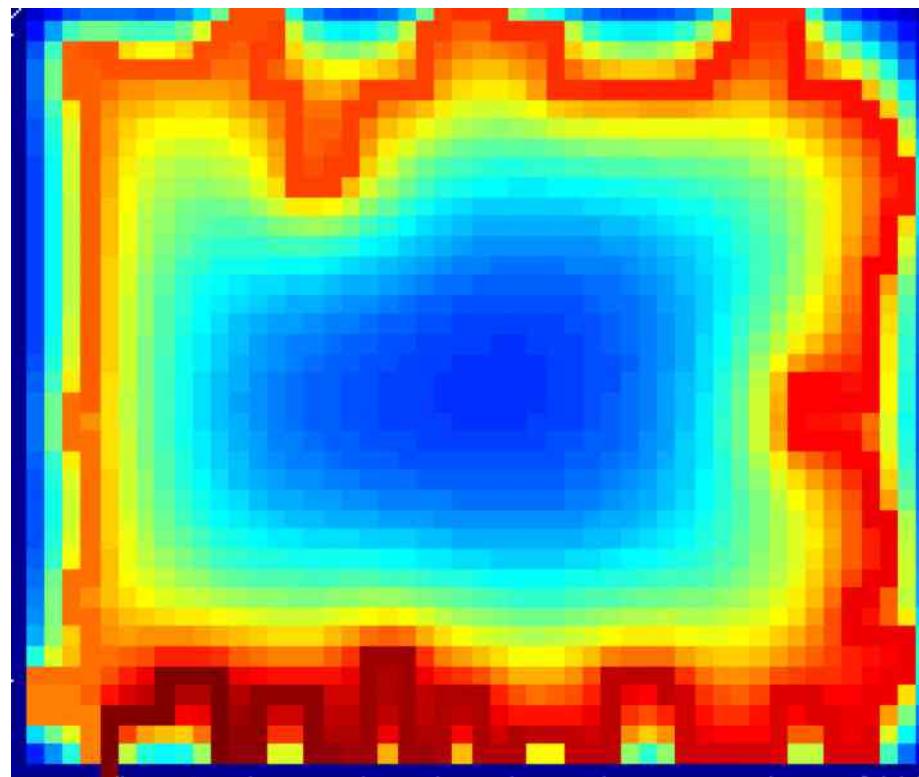


Abbildung 2: Start (nach wenigen Iterationen, $val > 300$)

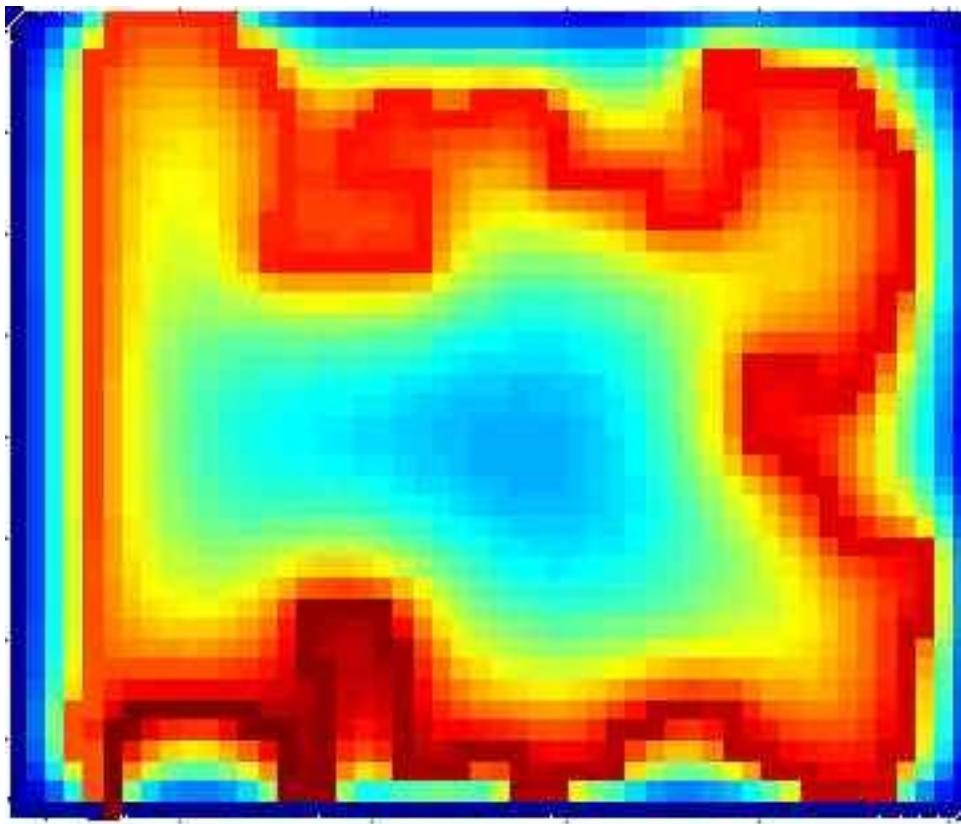


Abbildung 3: Start ($val = 90$)

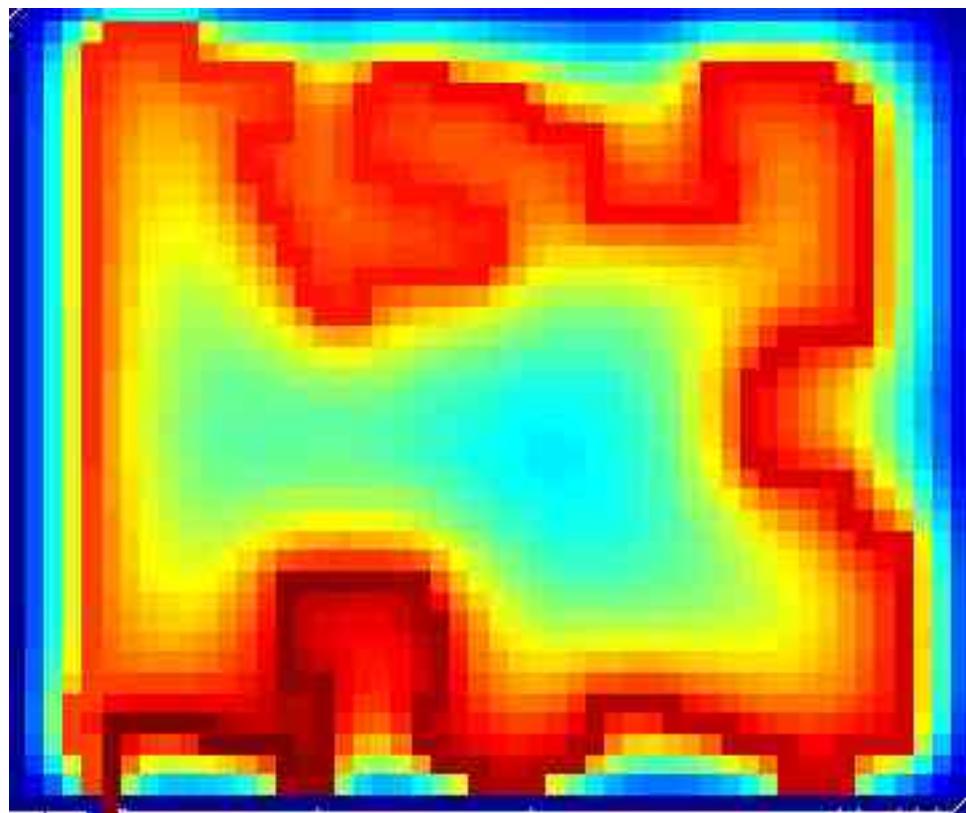


Abbildung 4: $val = 60$

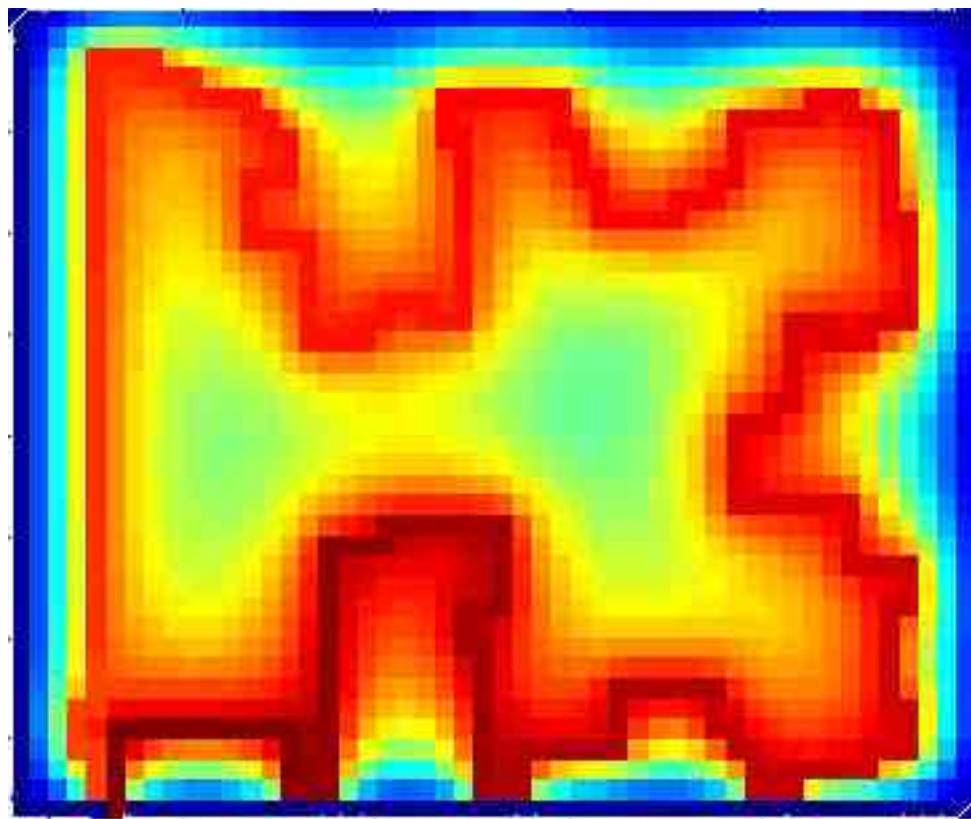


Abbildung 5: $val = 40$

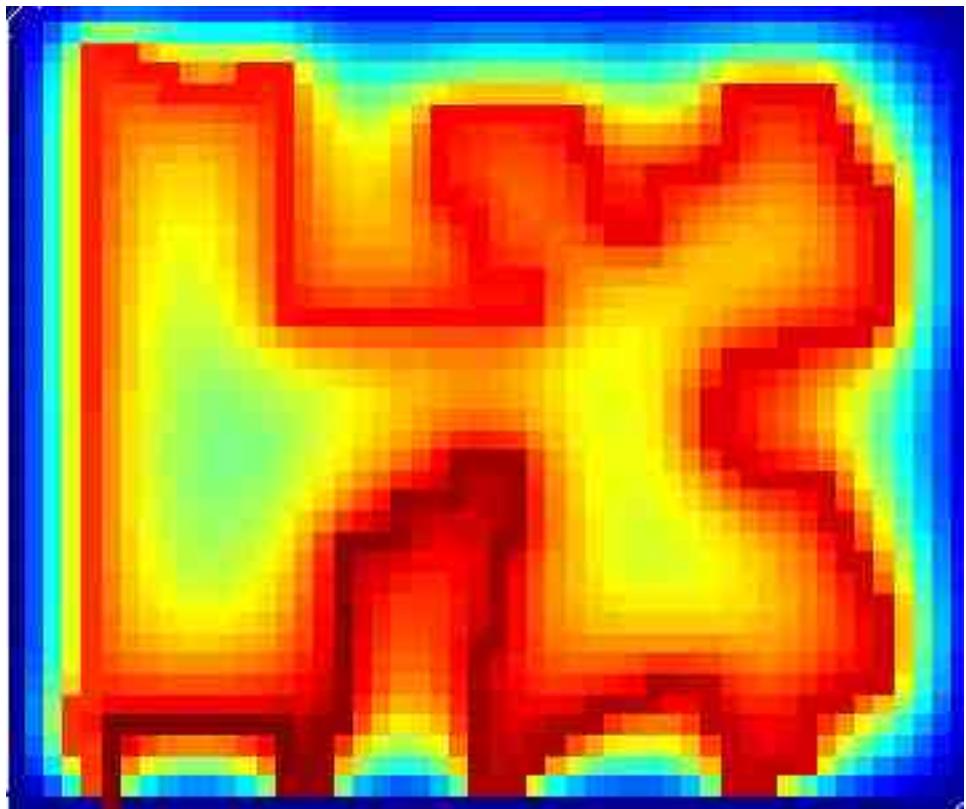


Abbildung 6: $val = 30$

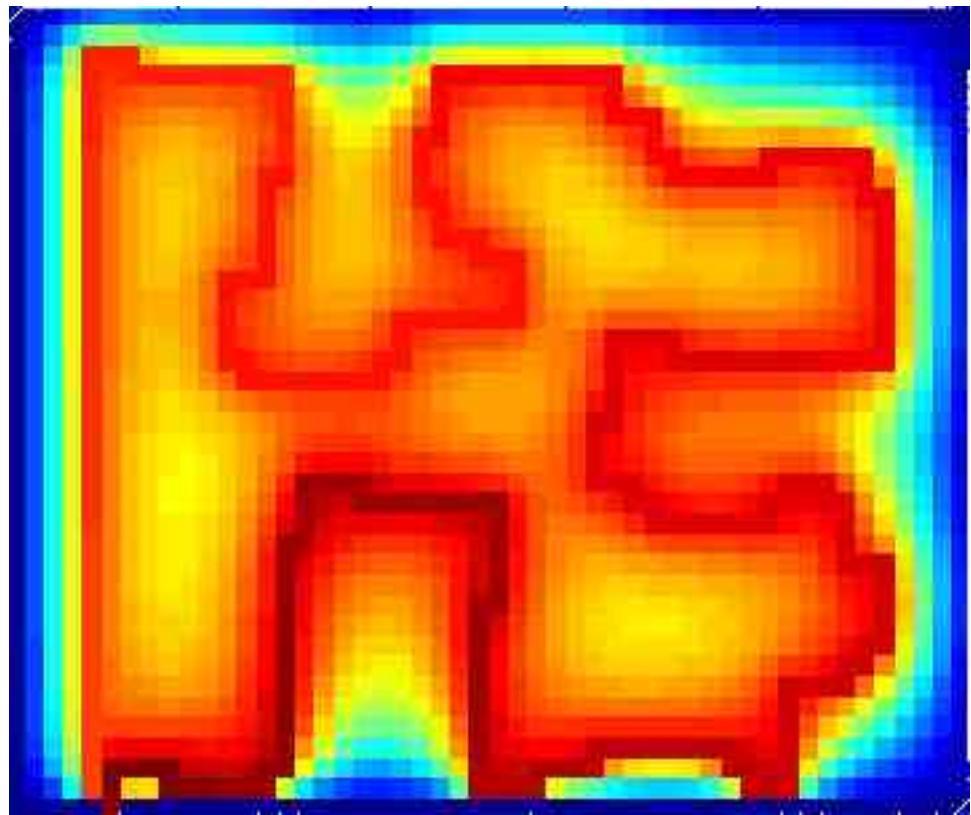


Abbildung 7: $val = 20$, offenbar Endpunkt

5.3 Entwicklungsschritte

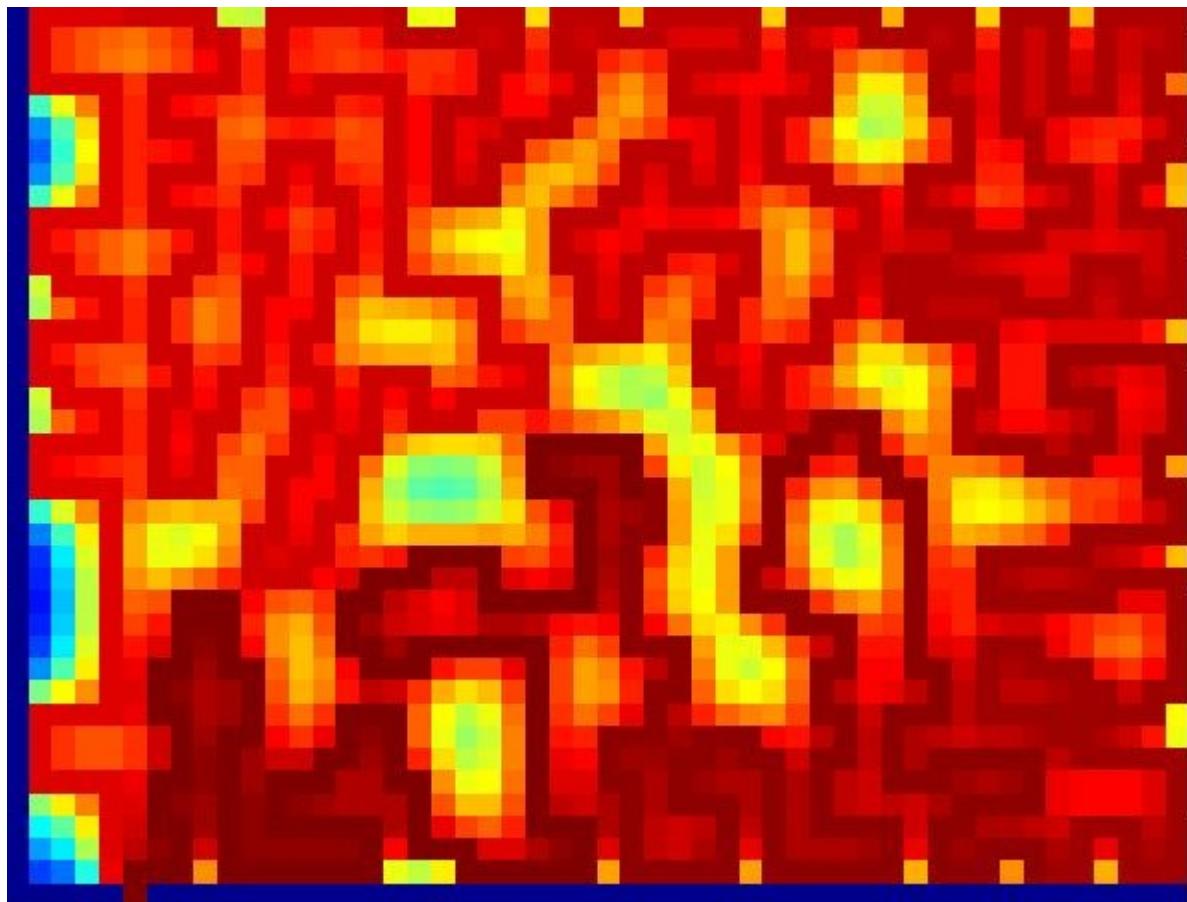


Abbildung 8: Einer der ersten halbwegs erfolgreichen Laufversuche

Wie Sie sehen können, ist hier die Länge der einzelnen Rohrstücke beileibe zu klein gewählt.

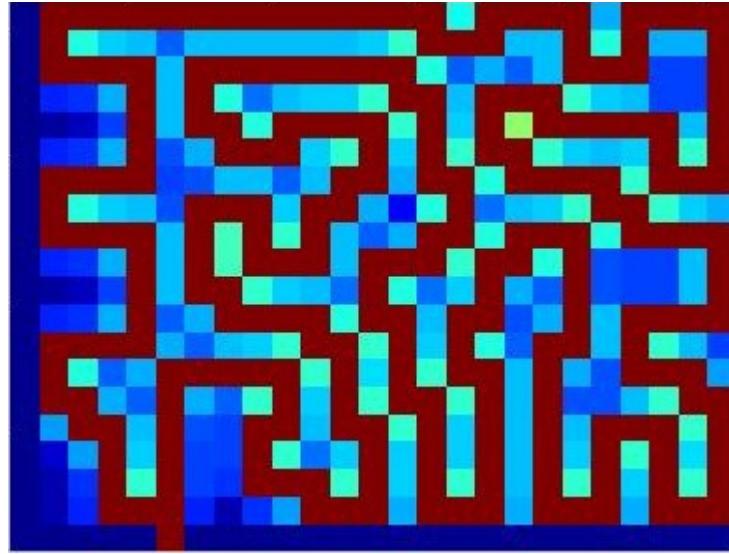


Abbildung 9: zu geringe Wärmeleitung eingestellt

6 Conclusio

6.1 Ergebnis

Aus den erhaltenen Ergebnissen lassen sich durchaus Parallelen zu den aktuell verlegten Fußbodenheizungen ziehen. Unsere Ergebnisse haben natürlich einige krumme Stellen mehr, aber die prinzipiellen geometrischen Muster sind annähernd dieselben die man sich bei einer modernen Fußbodenheizung erwartet. Offenbar ist das Optimum in der Nähe einer Art Heizschlange erreicht. Wenn man es jetzt schafft, die Parameter wirklich realistisch an die Umwelt anzupassen, kann man eventuell aus einigen Simulationsdurchläufen schlussfolgern, wie eng diese gelegt werden muss um eine optimale Heizwirkung zu erzielen. (In unserem Modellraum aus der Iterationsserie entsprechen offenbar drei Einstülpungen dem Optimum.

6.2 Ausbaumöglichkeiten

Das Modell ist allerdings noch ausbaufähig, wir mussten an einigen Parametern schrauben um sinnvolle Ergebnisse zu erhalten, da wir nur die Rohre, nicht aber den darüüber verlegten Fußboden und den Raum darüber simulie-

ren. Daher entsprechen die Parameter für die Wärmeleitfähigkeit des Bodens und den Wärmeverlust an die Raumluft leider noch keinen realistischen Werten. Ein erster Ansatz für eine Erweiterung des Projekts wäre daher auch das Einbauen von zusätzlichen Schichten - und damit der vernachlässigten dritten Dimension. Das erfordert natürlich auch einen weitaus größeren Rechenaufwand. Dies wiederum könnte man kompensieren indem man die stationäre Lösung der Wärmeleitgleichung anstatt der allgemeinen Lösung heranzieht und/oder bessere Verfahren verwendet.