

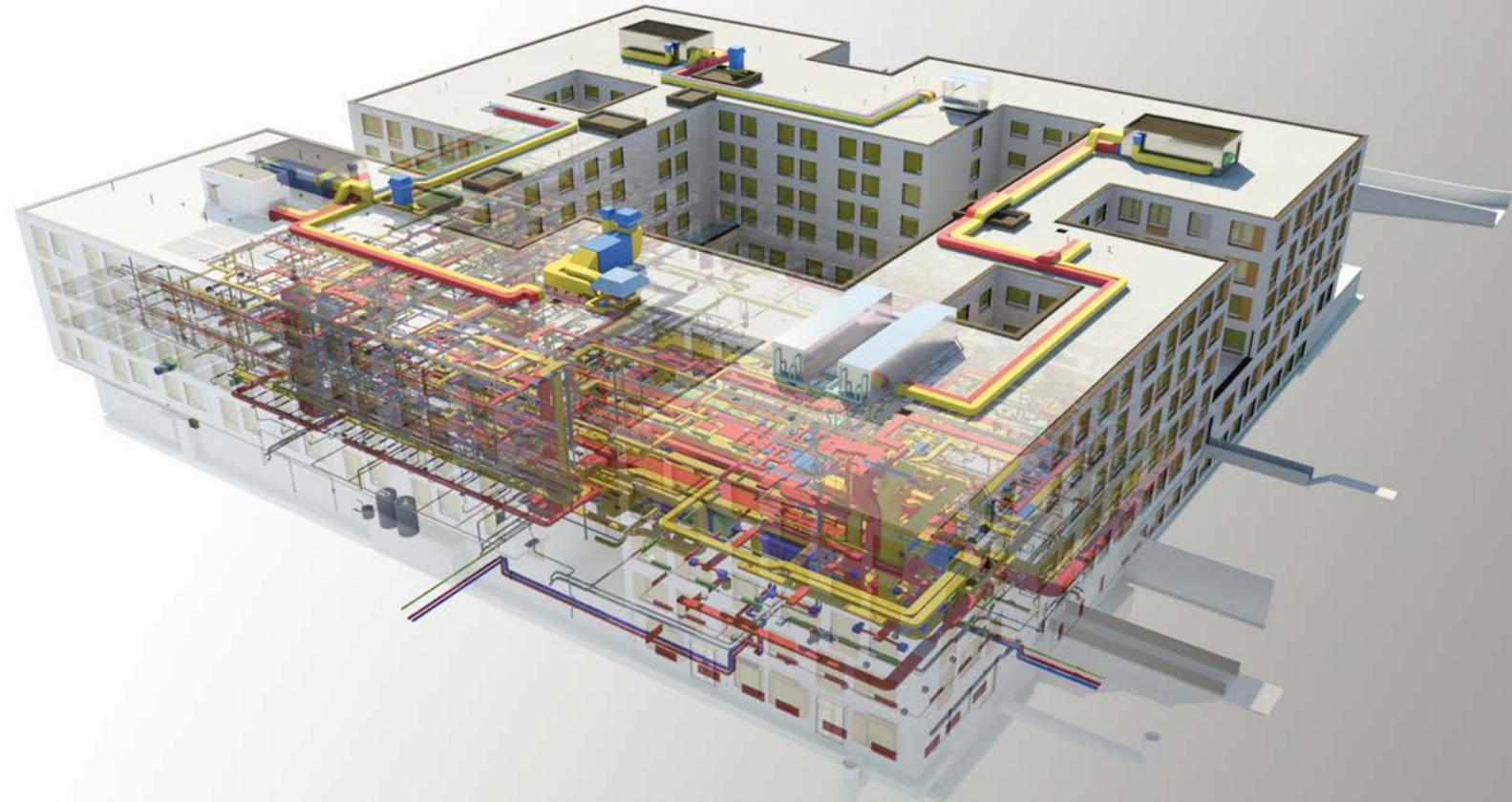
2017

Ernst & Sohn Special

November 2017, S. 132–135
A 61029

Sonderdruck

BIM - Building Information Modeling



Die Digitalisierung von Infrastrukturprojekten und ihre Herausforderungen

Vom Verhältnis zwischen Konstruktion und Numerik, von benutzerfreundlichen Lösungen und der Bedeutung eines erfahrenen Ingenieurs

Die Digitalisierung von Infrastrukturprojekten und ihre Herausforderungen

Vom Verhältnis zwischen Konstruktion und Numerik, von benutzerfreundlichen Lösungen und der Bedeutung eines erfahrenen Ingenieurs

Der Bausektor wird endlich digital und wagt damit einen Schritt in die Zukunft. 3D-Modelle sind da längst ein alter Hut, wir sprechen nun von digitalen Gebäudemodellen, die sämtliche Informationen enthalten. Um solche Modelle erstellen zu können, bedarf es des Einsatzes entsprechender Software. Egal, auf welchen Hersteller man setzt: Die Gebäudemodelle können sich sehen lassen, zumindest im Hochbau. Darauf, wie das im Infrastrukturbau aussieht, geht dieser Beitrag ausführlich ein.

Betrachtet man keine Gebäude mehr, sondern Ingenieurbauwerke, wird manchmal selbst ein schlichtes 3D-Modell zur Herausforderung. Die Erstellung von Freiformflächen ist zwar in fast allen Programmen machbar, aber das korrekte Darstellen einer Kuppe in der Gradienten- oder einer Klothoide in der Achse ist in den meisten BIM-fähigen Softwarelösungen noch nicht ohne entsprechende Hilfsmittel realisierbar. Folglich greifen wir hier oft auf die Software anderer Disziplinen zurück, beispielsweise auf Siemens NX aus dem Maschinenbau. Letztere bietet gute Möglichkeiten, Brücken mit Kuppe beziehungsweise Wanne und Übergangsbogen in kürzester Zeit zu erstellen und das Modell mit Informationen wie Material und Bauphasen zu verknüpfen. Zahlreiche Projekte haben wir so erfolgreich abgeschlossen und werden Siemens NX weiterhin einsetzen. Doch spätestens bei der 3D-Bewehrung stößt auch dieses Programm, wie jedes andere, an seine Grenzen.

– i –

Spaghetti Code und auf das Wesentliche zu reduzierende Erwartungen

Als innovatives Unternehmen möchten wir uns weder auf eine Software festlegen – zumal die Kooperation mit anderen Firmen stets eine gewisse Flexibilität in dieser Hinsicht erfordert –, noch uns von den Grenzen eines Programmes einschränken lassen. So suchen wir immer neue Möglichkeiten und Wege, unsere Modelle, als auch das Erstellen dieser Modelle effizienter zu machen. Bald fanden wir unseren Weg zur grafischen Programmieroberfläche Dynamo Studio. Diese grafische Entwicklungsumgebung bietet programmieraffinen Ingenieuren und Konstrukteuren eine effiziente Möglichkeit, ihre auf Achse und Gradienten basierenden Freiformflächen in eine BIM-fähige Software zu übergeben: Autodesk Revit. Auch auf diesem Weg haben wir bislang einige BIM-Projekte erfolgreich abgewickelt. Doch der sogenannte „Spaghetti-Code“ ist nicht jedermanns Sache – und wir können auch nicht von Ingenieuren und Konstrukteu-

Der sogenannte „Spaghetti-Code“ ist nicht jedermanns Sache – und wir können auch nicht von Ingenieuren und Konstrukteuren erwarten, sich mit den Grundsätzen der Programmierung auseinanderzusetzen, wenn sie eine Brücke konstruieren sollen.

```
5513     /// <summary>
5514     /// Creates ReferencePoint with the height of the given
5515     /// gradient on the given station of the alignment.
5516     /// </summary>
5517     /// <param name="uiApp">The active User Interface Application.</param>
5518     /// <param name="doc">The active document.</param>
5519     /// <param name="modelCrv">The curve that represents the alignment.</param>
5520     /// <param name="x">The station on which the calculation for the height is taken
5521     place.</param>
5522     /// <param name="z">The height of the gradient on the Station x.</param>
5523     /// <param name="endX">The length of the alignment part.</param>
5524     /// <returns>A ReferencePoint with the given height.</returns>
5525     public ReferencePoint PointWithHeight(UIApplication uiApp, Document doc, ModelCurve
5526     modelCrv, double x, double z, double endX)
5527     {
5528         //Find point on alignment
5529         PointLocationOnCurve LocOnCrv = new PointLocationOnCurve
5530         (PointOnCurveMeasurementType.SegmentLength, x,
5531         PointOnCurveMeasureFrom.Beginning);
5532         PointOnEdge poe = uiApp.Application.Create.NewPointOnEdge
5533         (modelCrv.GeometryCurve.Reference, LocOnCrv);
5534         ReferencePoint refPt = doc.FamilyCreate.NewReferencePoint(poe);
5535     }
```

Bild 1. C# Codeausschnitt

ren erwarten, sich mit den Grundsätzen der Programmierung auseinanderzusetzen, wenn sie eine Brücke konstruieren sollen.

Es musste endlich eine benutzerfreundlichere Lösung gefunden werden – wir stiegen dazu noch tiefer in Revit ein. Autodesk bietet mit der Revit API die Möglichkeit der Makro- und Add-in-Programmierung in einer NET-Framework-Umgebung. Makros und Add-ins haben den Vorteil, dass Benutzer den dahinterliegenden Code nicht sehen und man programmseitig die Eingangswerte über Windows-Eingabefenster von den Usern abfragen kann. Das schien uns einen Versuch wert. Die Windows-Eingabefenster waren für uns ein Grund, eine Add-in-Programmierung als eigenständiges Projekt zu starten. Doch wie bei jedem Projekt dieser Art mussten wir auch hier unsere Erwartungen, was das Add-in können sollte, auf das Wesentliche reduzieren.

– ii –

Sich summierende Fehler

Denn sobald man sich näher mit der Materie auseinandersetzt, wirft man Fragen auf, die mit der eigentlichen Konstruktion nichts mehr zu tun haben, sondern eher numerischer Natur sind. Wir sprechen oft von Genauigkeiten. Wie genau ist ein Programm? Kann es mir meine Absteckkoordinaten auf die fünfte Nachkommastelle ausgeben? Ist bei einem verschmolzenen Körper die Interpolation richtig und exakt? Was passiert mit den stetigen Funktionen, über die ich meine Trasse beschreibe? Wie bekomme ich tatsächlich den Übergangsbogen in das Programm? Die Antwort scheint simpel: Eine über Punkte definierte Kurve. Eine Klothoide

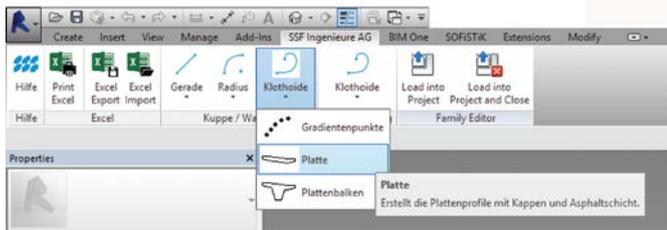


Bild 2. Revit-Multifunktionsleiste mit „Custom Ribbon Tab“

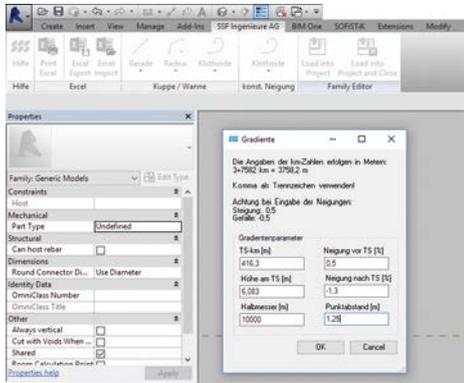


Bild 3. Eingabe der Gradiente in Revit mittels Windows-Eingabefenster

wird über eine stetige Funktion, die ein Integral ist, beschrieben. Damit ein Computer das umsetzen kann, müssen ausreichend diskrete Punkte über eine Reihenentwicklung berechnet werden, durch die eine interpolierte Kurve gelegt wird. Diese Kurve stellt dann, im Rahmen unserer Genauigkeiten, eine Klothoide dar. Das einzig Präzise an diesem Vorgehen ist die ursprüngliche mathematische stetige Funktion. Bereits die Reihenentwicklung stellt lediglich eine Annäherung, dar. Diese ist aber notwendig, um die entsprechenden Punkte durch den Computer berechnen zu lassen. Die Punkte beinhalten also schon eine gewisse Abweichung. Hinzu kommen Fehler, die darauf basieren, dass der Rechner nur Zahlen einer gewissen Länge abspeichern kann. In den meisten Fällen eine Dezimalzahl „double“, die eine auf 15 Stellen genaue Fließkommazahl darstellt. Der Fehler, den wir bei einer ordentlichen Approximation machen, ist also minimal; und dennoch ist er vorhanden, summiert sich mit jedem Schritt weiter auf.

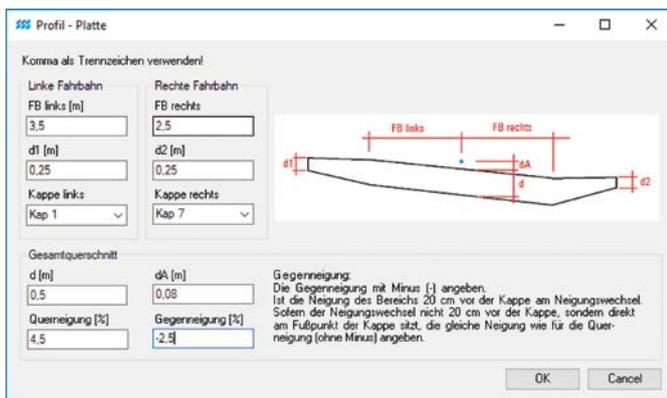


Bild 4. Eingabe der Querschnittswerte über Windows-Eingabefenster

– iii –

Relevanz des numerischen Ansatzes

All dessen müssen wir uns bewusst sein, wenn wir von Digitalisierung reden. Wir erstellen Modelle und Abbilder, die niemals eins zu eins der Realität entsprechen werden. Erst nähert man die Punkte der Achse an, dann legt man eine interpolierte Kurve durch diese Punkte. Auf dieser Kurve sucht man approximierte Punkte, denen anschließend berechnete Höhen vergeben werden, um wieder eine angenäherte Kurve durch diese Punkte zu legen, die die Gradienten darstellen soll. Wie oft haben wir diesen minimalen Fehler nun also aufsummiert? Ist der Fehler noch minimal genug? Wenn man Bautoleranzen betrachtet, die im Zentimeterbereich liegen, stellt sich die Frage, warum wir Absteckkoordinaten im Hunderttausendstel angeben. Das entspricht einem Mikrometer, also einem Tausendstel Millimeter. Ein gutes Bautachymeter hat eine Toleranz von ± 1 mm. Auch wenn wir bedenken, dass wir bei unseren ganzen Annäherungen Fehler machen, können wir davon ausgehen, dass jedes Programm, bei dem eine ordentliche Approximation hinterlegt wurde, eine Klothoide ausreichend exakt berechnen kann. Dies bedeutet, dass der numerische Ansatz, die Reihenentwicklung oder Approximation der stetigen Funktion, für die Genauigkeit unserer Berechnung viel relevanter ist als die Frage nach den Stellen, die ein Computer berechnen kann. Denn wer sein Modell auf einer grundlegend falschen Annahme aufbaut, wird auch bei tausend Stellen kein zufriedenstellendes Ergebnis erhalten.

– iv –

Faktor Mensch

Neben diesen numerischen Fragestellungen trifft man bei einem solchen Projekt auch früh auf konstruktive Fragen, die sonst erst wesentlich später gestellt werden. Wird der Flügel des Widerlagers im Radius geschalt? Oder schalt man nur den Kragarm und die Kappen im Radius und lässt den Flügel gerade? Lohnt es sich, Kragarm und Kappe bei einem Bogenstich von 0,5 mm im Radius zu konstruieren, auch wenn diese am Ende vermutlich gerade gebaut werden? Wer entscheidet, dass die Brücke gerade gebaut wird? Und wann? Und wer übernimmt die Verantwortung für diese Entscheidung? Während das Herausarbeiten dieser schwierigen konstruktiven Stellen in einer konventionellen Planung auch zu einem späteren Zeitpunkt noch erfolgen kann, fordert ein digitales Modell schon zu Beginn, dass man sich festlegt.

Digitalisierung im Bausektor kann also nur im Zusammenhang mit den dafür abgestimmten Bau- und Planungsprozessen sinnvoll erfolgen. In einem funktionierenden BIM-Prozess werden solche Entscheidungen frühzeitig getroffen und im BIM-Abwicklungsplan, dem BAP, dokumentiert. Damit dieser BIM-Prozess und die Digitalisierung reibungslos ablaufen, müssen wir unsere Herangehensweise grundlegend ändern. Dazu gehört, Probleme nicht im stillen Kämmerchen auszutüfteln, sondern diese offen anzusprechen; außerdem Verantwortung zu übernehmen und, zusammen mit anderen Planungsbeteiligten, gemeinsam zu agieren. Feedback-Management sollte ge-

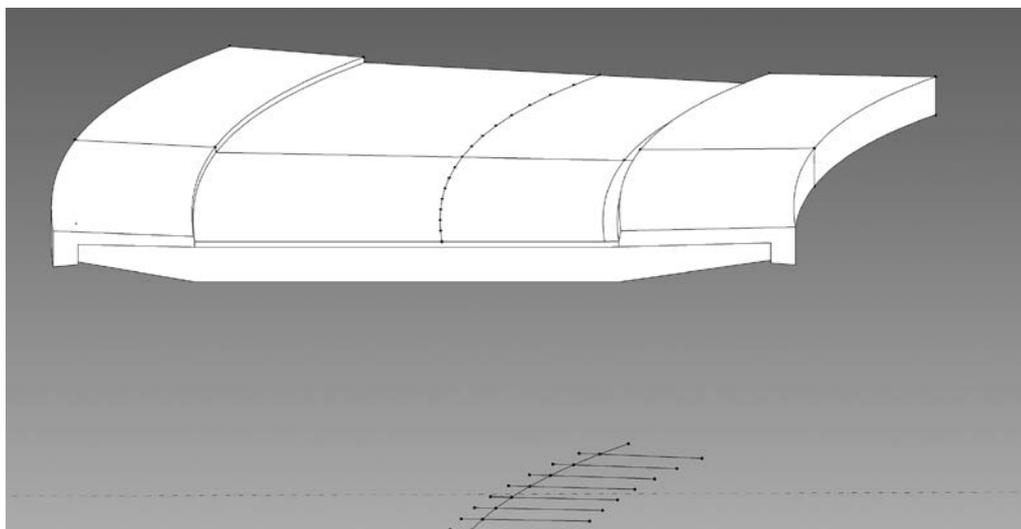


Bild 5. Fertiger Brückenüberbau aus automatisch generierten Querschnitten (Abb.: SSF-Ingenieure)

lebt werden, anstatt Schuldzuweisungen zu praktizieren. Wir alle müssen mit der Umsetzung neuer Prozesse im eigenen Unternehmen beginnen, wenn wir in Zukunft digital und effizient planen wollen.

Digitalisierung bedeutet in vielerlei Hinsicht auch Automatisierung. So stellt sich die Frage, ob es erforderlich ist, dass ohne Zutun des Konstrukteurs oder Ingenieurs rein programmtechnisch ein Brückenmodell erstellt wird. Darüber hinaus muss geklärt werden, ob sich sämtliche Fälle rein regelbasiert lösen lassen. Immer wird es Situationen geben, die eine gewisse Nacharbeit durch den Konstrukteur oder Ingenieur erfordern. Denn es lässt sich nicht alles – im kleinsten Detail mit tausend Ausnahmen – im Voraus mittels eines Programmes abdecken. Darüber hinaus ist Software nur so gut wie ihr Programmierer. Und dieser macht, wie jeder andere Mensch auch, seine Fehler. Durch unsere digitale Umwelt neigen wir immer mehr dazu, unseren Computern blind zu vertrauen. Aber wie im numerischen Ansatz können sich auch im konstruktiven falsche Lösungen einschleichen. Es bedarf also immer einer abschließenden Überprüfung und Nacharbeit des Modells, nicht nur durch regelbasierte Software, sondern auch durch einen erfahrenen Ingenieur oder Konstrukteur. Unser Berufsstand bleibt trotz der fortschreitenden Automatisierung unverzichtbarer Bestandteil der Wertschöpfungskette „Bau“.

Vor diesem Hintergrund und unter Berücksichtigung einer ganzen Palette von Unsicherheitsfaktoren haben wir unser Add-in programmiert.

– V –

Add-in deckt niemals sämtliche Fälle ab

Wir können nun beliebige Querschnitte an die entsprechenden Stellen mit der berechneten Höhe platzieren und

so ohne weitere Hilfsmittel unseren Brückenkörper zielgenau erstellen. Die Eingabe der Ausgangswerte erfolgt über Windows-Eingabefenster, die nacheinander die notwendigen Werte abfragen. Das Einzige, was der Benutzer braucht, sind Trassendaten und Querschnittsabmessungen. Sollte ein benutzerdefinierter Querschnitt verwendet werden, können auch nur die Gradientenpunkte ausgegeben werden und die eigenen Querschnitte händisch platziert werden. Dieser Überbau kann anschließend in Revit auch 3D-bewehrt werden: Sämtliche Schal- und Bewehrungspläne können so direkt aus Revit generiert werden, ohne dass in AutoCAD eine Nacharbeit der Pläne erforderlich wird. Doch eine Brücke besteht aus mehr als nur dem Überbau. Weitere Module sind denkbar – die Erstellung der Böschung und des Widerlagers, Schutzeinrichtungen, Pfeiler und mehr. Hier gilt ebenfalls, dass wir mit unserem Add-in niemals sämtliche Fälle abdecken können, wir stellen diesen Anspruch erst gar nicht. Wir bieten mit unserem Add-in unseren Ingenieuren und Konstrukteuren nur eine weitere Möglichkeit von vielen, um eine Brücke oder andere Ingenieurbauwerke in 3D zu konstruieren.

Und auch wenn wir bei SSF Ingenieure gerne weiterhin die Grenzen unserer Software übertreten und eine mitgestaltete digitale Zukunft kaum erwarten können: Wir würden uns über eine zügige Hilfestellung durch die großen Softwarehäuser freuen. Damit wir uns wieder auf das Wesentliche konzentrieren können: das Konstruieren von Brücken und Tunneln!

*Dipl.-Ing. Dietrich Sundmacher, SSF Ingenieure AG;
Helene Meyer, M.Sc., SSF Ingenieure AG*

www.ssf-ing.de