Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Eidgenössisches Departement für Umwelt, Verkehr, Energie und Kommunikation UVEK

Bundesamt für Energie BFE
Energieforschung

Schlussbericht vom 17.12.2019

OPTIVITRAGE - Optimisation du choix des éléments vitrés dans la construction

Optimization of the choice of glazed elements in building constructions

Datum: 17.12.2019
Ort: Bern

Auftraggeberin:
Bundesamt für Energie BFE
Energieforschung
CH-3003 Bern
www.bfe.admin.ch

Auftragnehmer/in:
Berner Fachhochschule
Architektur, Holz und Bau
Solothurnstrasse 102
CH-2504 Biel-Bienne
www.ahb.bfh.ch

e4Tech Software SA
Av. Juste-Olivier 2,
CH-1006 Lausanne
www.e4tech-software.com

Autor/in:
Driton Komani, Berner Fachhochschule Architektur, Holz und Bau, driton.komani@bfh.ch
Flavio Foradini, e4Tech Software SA, flavio.foradini@e4tech.com

BFE-Bereichsleitung: Rolf Moser, Enerconom SA, moser@enerconom.ch
BFE-Programmleitung: Andreas Eckmanns, Bundesamt für Energie BFE
BFE-Vertragsnummer: SI/501065-01

Für den Inhalt und die Schlussfolgerungen sind ausschliesslich die Autoren dieses Berichts verantwortlich.

Titelbild: Wohnhaus mit optimierter Verglasung

Bundesamt für Energie BFE
Mühlestrasse 4, CH-3063 Ittigen; Postadresse: CH-3003 Bern
Tel. +41 58 462 56 11 · Fax +41 58 463 25 00 · contact@bfe.admin.ch · www.bfe.admin.ch

Zusammenfassung

Ziel des Projektes war die Entwicklung eines Werkzeugs zur Bestimmung der optimalen Eigenschaften und Abmessungen von Verglasungen von Gebäuden in Abhängigkeit von einzelnen Bauvorhaben auf Basis der SIA380/1 und SIA 2044. Das Tool soll in die bekannte Software *Lesosai*, einer Software zur Bestimmung der Energiebilanz von Gebäuden, integriert werden.

Gemäss SIA 380/1 wird die Energiebilanz monatlich berechnet und zu einem jährlichen Energieverlust oder -gewinn addiert. Für das Wohnen ist aber nicht nur die Energiebilanz wichtig, auch der Komfort und das Raumklima sind wesentliche Faktoren für ein komfortables und nachhaltiges Wohngebäude. Diese beiden Faktoren sind in der SIA 2044 definiert.

Eine Fallstudie wurde durchgeführt, um die Bedeutung von verglasten Elementen in der Gebäudehülle und das große Potenzial zur Senkung des Energieverbrauchs durch eine Erhöhung der Solarenergiegewinne zu veranschaulichen. Die Gebäudehülle ist als Einheit zu betrachten und ihr Design sollte bereits in der Planungsphase eines Bauprojekts optimiert werden. Die Fallstudie zeigte, dass vollverglaste Fassaden aus energetischer Sicht nicht unbedingt die optimale Lösung sind, da sie in der warmen Jahreszeit zu Überhitzung und erhöhtem Energieverbrauch durch Klimatisierung führen können.

Allerdings ist es schwierig, den besten Parameter-Mix, basierend auf der Energiebilanzberechnung SIA 380/1, hinsichtlich der energetischen und wirtschaftlichen Sichtweise zu finden.

Mit einem Standardansatz in der Software erfordert dies eine sehr komplexe Berechnung, die zu viel Zeit und Rechnerleistung in Anspruch nehmen würde, um alle möglichen Kombinationen zu berechnen. Das Projektteam fand und implementierte eine Lösung für eine sogenannte Vorhersagefunktion, die auf einer mathematischen Wahrscheinlichkeitstheorie nach dem Vorbild der Gaußschen Prozesse basiert. Diese Vorhersagefunktion wurde in einem *Mathlab*-Code implementiert, der dann in den *Lesosai*-Code *Delphi* übersetzt wurde. Viel Arbeit wurde in die Überprüfung der Korrektheit dieser Implementierung investiert.

Die Ergebnisse des Projekts wurden in *Lesosai* umgesetzt. Dem Planer oder Architekten stehen damit die Daten der optimalen Oberfläche und der Eigenschaften der Verglasung zur Verfügung.

Résumé

L'objectif du projet était le développement d'un outil permettant de déterminer les caractéristiques et dimensions optimales des panneaux de vitrage dans les bâtiments, en fonction de projets de construction individuels, sur la base des normes SIA380/1 et SIA 2044. L'outil devra être intégré dans *Lesosai*, un logiciel bien connu pour déterminer le bilan énergétique des bâtiments.

Grâce à la norme SIA 380/1, le bilan énergétique est calculé mensuellement et additionné à une perte ou gain énergétique annuel. Pour le logement, non seulement le bilan énergétique est important, mais le confort et la climatisation sont également des facteurs essentiels pour un bâtiment résidentiel confortable et durable. Ces deux facteurs sont définis dans la SIA 2044.

Une étude de cas a été réalisée pour illustrer l'importance des éléments vitrés dans l'enveloppe du bâtiment ainsi que le grand potentiel de réduction de la consommation énergétiques par une augmentation des gains en énergie solaire. L'enveloppe du bâtiment doit être considérée comme une entité à part entière et sa conception doit être optimisée pendant la phase de planification d'un projet de construction. L'étude de cas a montré que les façades entièrement vitrées ne sont pas nécessairement la solution optimale du point de vue énergétique, car elles peuvent entraîner une surchauffe pendant la saison chaude et augmenter ainsi la consommation d'énergie due à la climatisation.

Cependant, il est difficile de trouver la meilleure combinaison des paramètres concernant les aspects énergétique et économique, en se basant sur le calcul du bilan énergétique de la SIA 380/1.

Avec une approche standard dans le logiciel, cela nécessite un calcul très complexe qui prendrait trop de temps pour évaluer toutes les combinaisons possibles. L'équipe du projet a trouvé et mis en œuvre une solution sur une fonction dite de prédiction, basée sur la théorie mathématique des probabilités modélisée sur des processus gaussiens. Cette fonction de prédiction fut implémentée dans un code *Mathlab*, qui fut ensuite traduit dans le code *Delphi* du logiciel *Lesosai*. Beaucoup de travail a été consacré à tester l'exactitude de cette mise en œuvre.

Les résultats du projet ont été mis en œuvre dans *Lesosai*. Ainsi, l'ingénieur planificateur ou l'architecte a à sa disposition les données de surface optimale et les caractéristiques du vitrage.

Summary

The objective of the project was the development of a tool to determine the optimal characteristics and dimensions of glazing panels in buildings in function of individual building projects on the bases of SIA380/1 and SIA 2044. The tool should be integrated in *Lesosai*, a well-known software for determining energy balances of buildings.

Due to SIA 380/1 the energy balance is calculated monthly and added up to a yearly energy loss or gain. For housing not only the energy balance is important, but the comfort and the climatization are also essential factors for a comfortable and sustainable residential building. These two factors are defined in the SIA 2044.

A case study was carried out to illustrate the importance of glazed elements in the building skin and the great potential to decrease energy consumption through an increase in solar energy gains. The building skin has to be considered as an entity and its design should be optimized during the planning phase of a construction project. The case study showed that fully glazed facades are not necessarily the optimal solution from an energetic point of view as it could lead to overheating during the warm season and cause increased energy consumption due to air-conditioning.

However, finding the best parameter mix, based on the SIA 380/1 energy balance calculation, regarding the energetical and economical view is difficult.

With a standard approach in the software it requires a very complex calculation which would take too much time to calculate all possible combinations. The project team found and implemented a solution on a so-called prediction function, which is based on mathematical probability theory modelled on Gaussian processes. This prediction function was implemented in a *Mathlab* code, which has been then translated in the *Lesosai* code *Delphi*. A big deal of work has been devoted into testing the correctness of that implementation.

The results of the project have been implemented in *Lesosai*. Thus, the planning engineer or architect has the data of optimal surface and characteristics of glazing at his disposal.

Anhang

Presentation of Flavio Foradini (E4tech Software SA.) at the conference "windays 2019" in Biel/Bienne on 29th March 2019.

Inhaltsverzeichnis

Abkürzungsverzeichnis

EPFL	Ecole Polytechnique Fédérale de Lausanne
BFH	Berner Fachhochschule
PVC window	Polyvinylchlorid-window
RBF	Radial Basis Function

1 The Setting and Goals

1.1 Introduction

The company *EPFL (Ecole Polytechnique Fédérale de Lausanne)* developed in 1984 the building physics software *Lesosai*. Since 2001 this software is developed and managed from the company *E4tech Software SA* based in Lausanne, Switzerland. This enables soft- ware engineers and building physicist to calculate the energy balance of a building and based on these data, to apply for a Minergie certificate or similar. The energy balance calculation is based on the SIA 380/1 [2], where all parameters and limit values are defined and described. The energy balance is calculated monthly and added up to a yearly energy loss or gain.
For housing not only the energy balance is important, but the comfort and the climatization are also essential factors for a comfortable and sustainable residential building. These two factors are defined in the SIA 2044 [5]. The hourly calculation based on the SIA 2044 makes it necessary to have an enormous processing power for the calculation of all the parameters needed.
Due to this fact the *Lesosai* developer implemented a prediction function denoted by p in their software, which permits to calculate the energy balance values much faster. The prediction function p has been implemented in a test version in the *Lesosai* software. We will choose a probabilistic point of view of our problem in order to achieve a fast calculating prediction function p. A necessary part of this project is to test the prediction function p wether it runs without any bugs or errors.
As a next step the *E4tech* team wants to upgrade the software with another feature, which is very helpful especially engineers and building physicists. With this application the user is able to find the best parameter mix by testing various combinations of the parameters to achieve their goals.

Finding the best parameter mix, based on the SIA 380/1 energy balance calculation, regarding the energetical and economical view is not easy, especially for inexperienced engineers and building physicists. Considering that fact the *E4Tech* team wants to upgrade their well-known software *Lesosai* with a new and very handy feature. This application gives the user the possibility to vary different parameters and calculate the energy balance of a building in planning. The application gives them the best design. Due to the fact that it is a very complex calculation the software would take too much time to calculate all possible combinations. Therefore, the *E4Tech* team has implemented a prediction function p, which is based on mathematical probability theory modelled on Gaussian processes.
The prediction function p is implemented in a Mathlab code, which has been translated in the *Lesosai* code *Delphi*. A big deal of work is devoted into testing the correctness of that implementation. The results show that the implementation of the so-called prediction function is done accurately and that the pred function itself is well implemented in *Delphi* and works fine.

1.2 Basics of energy balance of buildings

In recent years, glazed components in the building skin have gained more and more importance in modern architectural design. Buildings with large windows or completely glazed facades are no rarities in today's built environment any more. Increased demand for large glazed components with low thermoconductive properties has led to important research efforts and to modern windows with improved structural and thermal performances.

At the same time, our society faces new challenges with an energy turnaround that aims at reducing the consumption of fossil fuels including nuclear energy. Given that the energy consumption for heating constitutes a major part of the total energy consumption in many developed countries – heating represents more than 30% of the total energy consumption in Switzerland [1] – new research efforts are needed to help reducing the energy need of contemporary households. It is therefore important to consider not only the technical aspects of the facade components, as for instance the windows, but also the building skin as an entity. In addition to other energy-saving strategies, the improvement of the facade design of new buildings from an energetic point of view is a promising approach to contribute to the energy turnaround challenge. Well-designed building skins with strategically located glazed elements improve considerable the gained solar energy and reduce therefore the needed heating energy.

In the next section we present a case study showing the influence of various building skin components on the thermal performance of a reference building. The main objective is the illustration of the order of possible energy savings due to an energetically improved design of the building. Further, we present an optimization strategy that allows finding an optimal facade design with respect to the energy consumption for heating. Integrated into a software tool, this approach provides engineers early on during a project with the possibility of finding a better building skin design, which means that they are able to determine building skin parameters that generate the best energy balance possible.

The energy consumption for heating of a building depends on various factors such as the geographic location, surrounding topography and buildings, orientation and the shape and size of the building, building insulation, thermal capacity of the building, transmission losses, light transmission, summer heat protection, heating and cooling energy requirements, energy needs for lighting, and the sources of energy used. The energy needed for heating is generally computed by determining the energy balance of the building including energy losses and gains, which is given by

$$f = (Q_T + Q_V) - \eta_g (Q_I + Q_S)$$

where $f = Q_h$ is the required annual energy for heating, Q_T is the energy loss by transmission, Q_V the energy loss by ventilation, η_g the energy conversion efficiency, Q_I the internal energy gain, and Q_S the solar energy gain. The quantities $Q_T, Q_V, \eta_g, Q_I, Q_S$ depend on a high number of parameters. Hence the energy balance $f = Q_h$ can be calculated depending on these parameters such as for example

$x_1 =$ *(area of windows north)*

$x_2 =$ *(area of windows south)*

$x_3 =$ *(shade coefficient north)*

$x_4 =$ *(shade coefficient south)*

$x_5 =$ *(g-value windows north)*

$x_6 =$ *(g-value windows south).*

$\vdots$ $\vdots$

$x_k =$ *(last parameter involved)*

For small buildings in planning the number of parameters $k \in \mathbb{N}$ can get already very high such as $k = 60$ or $k = 100$. This adds to the complexity of finding the optimal building skin design $(x_1, \dots, x_k) \in \mathbb{R}^k$. We are looking the **optimal building skin design** which by definition is a vector $(x_1, \dots, x_k) \in \mathbb{R}^k$ that minimizes the **energy balance function**

$$f: \mathbb{R}^k \rightarrow \mathbb{R}$$

This function has been computer simulated by *E4Tech* within the framework of the software *Lesosai*. The aim of any civil engineer is to find the best parameter design $(x_1, \dots, x_k)$ for which the energy balance $f(x_1, \dots, x_k)$ is minimal. To do so one has to insert all possible vectors $(x_1, \dots, x_k)$ and calculate the value $f(x_1, \dots, x_k)$ and pick the vector that gives the minimal energy balance. However, there is a Problem: The calculation of the energy balance $f(x_1, \dots, x_k)$ performed in *Lesosai* is **time expensive**. That's why we need to have a strategy in *Lesosai* that delivers an optimal building skin design in reasonable time.

1.3 Thermal performance of buildings: a case study

The various losses and gains have multiple causes. For instance, the wall, the windows and the roof contribute to the loss by transmission, and various household appliances, the lighting and the room occupation add to the internal energy gain.

The energy losses and gains of this case study are computed following the guidelines of the SIA 380/1 norm [2]. A detailed description is beyond the objective of this work.

The potential of an improved building skin for decreasing the heating energy consumption of residential houses is illustrated in this case study. The reference building is a typical single-family home with a first floor built in masonry and a second floor with a wooden structure. It is located in Lajoux in Switzerland at an altitude of 1020m above sea level. The principal properties are summarized in Table 1.

Properties	**Value**
Living area (reference area) [m^2]	225.6
Area of building skin [m^2]	451.2
Area of vertical facade [m^2]	214
Area of glazed elements [m^2]	31.8
Area of south-oriented facade [m^2]	70
Area of glazed elements in south-oriented facade [m^2]	20
Shade coefficient of south-oriented facade	0.77

Table 1: Properties of reference building

It can be expected that the type of window and glasses, the insulating property of the wall, and the share of the glazed area in the south-oriented facade are some of the most influential factors for the energy balance of the building. In addition, the location of the building is of great importance for the heating energy consumption as well as for the energy-saving potential of clever building skin designs. The influence of these main factors is here analysed by computing the energy balance for the reference building with two different window types, two different glazing elements, two different areas of glazed elements in the south-oriented facade, two different types of wall insulation, and at two different locations. In total, this results in 32 different cases, which are then compared with respect to their need of heating energy.

The two different types of windows used in this case study are summarized in Table 2.

Description	Conductance of frame [W/m²K]	Area share of glass
PVC window	1.45	0.79
Wood/Aluminum window	1.46	0.88

Table 2: Properties of windows used in the case study

Description	Conductance [W/m²K]	Psi value [W/mK]	Energetic transmission
Insulated double-glazed	1.1	0.05	0.6
Insulated triple-glazed	0.6	0.04	0.5

Table 3: Properties of window glasses used in the case study

Furthermore, the share of glazed elements in the south-oriented facade is modified from 28%, which corresponds to a facade of a typical family home, to 45%, where the entire top floor is equipped with glazed elements. The two different types of wall insulations present conductances of 0.14 W/m²K and 0.06 W/m²K. The energy balance is computed for the reference building at two locations. The first location, Lajoux, is in a mountainous region at an altitude of 1020m above sea level, whereas the second location, Neuchâtel, is at a much lower altitude (480m).

1.4 Results and the discussion of the study case

The results of the energy balance computation of the reference building at Lajoux are shown in Figure 1.

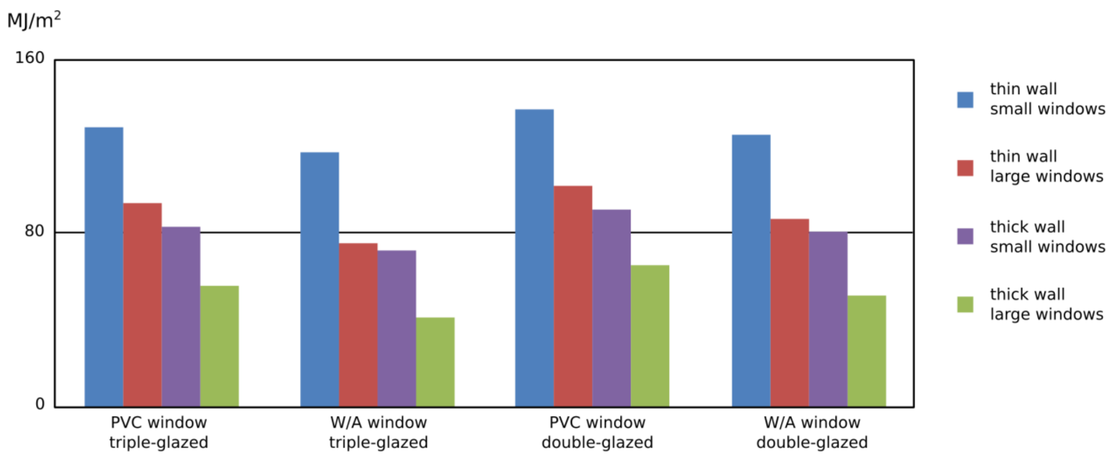


Figure 1: Required annual heating energy for reference building at Lajoux

As expected, the wood/aluminium windows (in this particular case) and the triple-glazed windows result, compared to their counterparts, in slightly lower energy needs for heating. This is due to lower conductance values and lower share of the frame area, which lead both to lower energy losses through the windows. However, the effect of both factors, the window type and the glass type, is considerably smaller than the energy saving achieved with more insulated (thicker) walls and larger shares of the south-oriented glazed area.

Similar results are observed for the energy balance of the reference building located in Neuchatel, as shown by Figure 2. In both cases, the energy savings with more insulated walls and larger south-oriented windows are essential and can reach values up to 50% of the total heating energy consumption of a household.

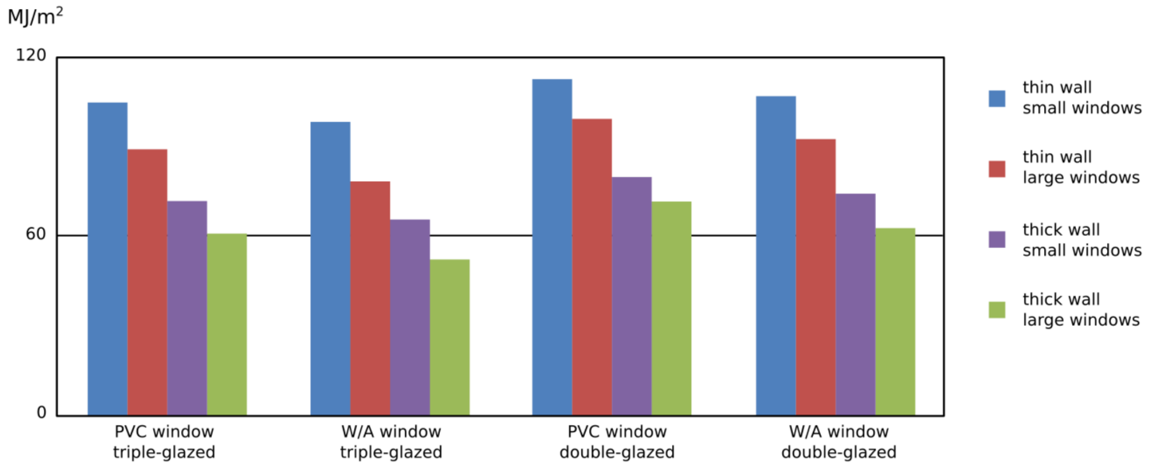


Figure 2: Required annual heating energy for reference building at Neuchâtel

This case study illustrates the importance of glazed elements in the building skin and the great potential to decrease energy consumption through an increase in solar energy gains. However, several aspects, which complexify the problem of finding an optimal building skin design for minimal energy consumption, have been neglected so far. First of all, the presented energy balance computations are based on a monthly estimated of energy losses and gains. In reality, however, large temperature variations occur within the duration of a month. The interaction of these variations and the thermal inertia of the building make the problem nonlinear and the computation of the annual energy consumption more challenging.

Moreover, one could think that a building with a fully glazed south-oriented facade and without any windows in the north-oriented facade is the optimal solution. This approach, however, would most probably lead to overheated buildings during the summer months, which then would require another energy-consuming technology: air-conditioning. It is therefore likely that the optimal building skin design with respect to minimizing heating/cooling energy needs is not an extreme case (*e.g.*, fully glazed facade). Finding such a (near-)optimal design is not a trivial problem, considering its nonlinear nature, and needs well adapted methods. One such strategy is presented in the following section.

It is also worth noting that an optimal building skin design resulting in minimal energy consumption might not necessarily be "the" optimal solution if other aspects, such as living comfort, structural feasibility, and construction cost are considered.

1.5 Conclusion

We have shown that glazed elements in the building skin have an essential effect on the energy balance of the building. It was illustrated that windows in the south-oriented facade can contribute favourably to the objective of decreasing energy consumption for heating by gaining additional solar energy. In order to be able to benefit from the solar energy, the building skin has to be considered as an entity and its design should be optimized during the planning phase of a construction project. We further highlighted that fully glazed facades are not necessarily the optimal solution from an energetic point of view as it could lead to overheating during the warm season and cause increased energy consumption due to air-conditioning.

1.6 Goal and strategy

We finally propose to start the optimization procedure with sampling plans of the type of latin hypercubes and analysed the potential improvement of their space-filling character with an evolutionary process as explained further in the chapter on mathematical methods.

2 The Mathematical Methods

2.1 Mathematical Setting

The SIA 380/1 norm proposes a function $f: \mathbb{R}^k \to \mathbb{R}$ being the annual energy of a building in planning. The input vectors $x = (x_1, \dots, x_k) \in \mathbb{R}^k$ mainly consist of building skin parameters. The energy balance $f: \mathbb{R}^k \to \mathbb{R}$ has been implemented and computer simulated in the software Lesosai. This simulation follows the calculations suggested by the SIA 380/1 norm. The calculation of the simulated energy balance $y = f(x_1, \dots, x_n)$ is time expensive which makes it a hard task to find the input vector that minimizes the energy balance function $f(x_1, \dots, x_n)$. We want to minimize the computer simulated energy balance $f(x)$ over all possible input vectors $x = (x_1, \dots, x_k) \in \mathbb{R}^k$ knowing the observed values

$$y_1 = f(x_1),\ y_2 = f(x_2), \dots,\ y_n = f(x_n)$$

at any sample set containing input vectors $\{x_1, \dots, x_n\} \subset \mathbb{R}^k$ where any input vector in the sample set $x_i = (x_{i1}, \dots, x_{ik}) \in \mathbb{R}^k$ consists of k building skin parameters. In order to minimize the function $f: \mathbb{R}^k \to \mathbb{R}$ we model it as a Gaussian process and extract some prediction function $p: \mathbb{R}^k \to \mathbb{R}$ which approximates the energy function f given the observed values $\{y_1, \dots, y_n\}$. We present an algorithm that will deliver the prediction function $p \approx f$ that approximates the energy balance. The advantage of p is that it is not time expensive to calculate the approximated energy balance values $p(x)$.

2.2 Gaussian Processes

Gaussian processes are studied very well and hence many codes and implementations already exist. Gaussian processes will enable us to come up with a prediction function $p: \mathbb{R}^k \to \mathbb{R}$ and quantify the uncertainty of our prediction, which is a desirable measure of uncertainty. We view the annual energy balance f as a **Gaussian process** being a function of the type

$$f: \mathbb{R}^k \times \Omega \to \mathbb{R}$$

meaning that the random vector $\big(f(x_1), f(x_2), \dots, f(x_n)\big)$ is Gaussian for any choice of input vectors $\{x_1, \dots, x_n\} \subset \mathbb{R}^k$. Here Ω is some non-specified probability space and $f(x_i): \Omega \to \mathbb{R}$ is interpreted as a random variable. Hence the computer simulated function is viewed as a random output $f(x_i) = f(x_i, \omega)$ for some elementary event $\omega \in \Omega$. For a fixed event $\omega_0 \in \Omega$ the function $x \mapsto f(x, \omega_0)$ is called **realization** of f. The computer simulated energy balance function $f: \mathbb{R}^k \to \mathbb{R}$ is seen as a random phenomenon yielding functions $x \mapsto f(x, \omega_0)$ as realisations. The important thing about Gaussian processes is that they are characterized by its mean and covariance functions. The **mean function** of a Gaussian process is a function $\mu: \mathbb{R}^k \to \mathbb{R}$ defined by the expectation value of the random variable $f(x): \Omega \to \mathbb{R}$ denoted by $\mu(x) = \mathbb{E}\big(f(x)\big) \in \mathbb{R}$. A finite set of input vectors $\{x_1, \dots, x_n\} \subset \mathbb{R}^k$ is called **sample plan**. A sample plan defines a random vector $X = \big(f(x_1), f(x_2), \dots, f(x_n)\big)$ with mean vector $\mu = \mathbb{E}(X) \in \mathbb{R}^n$. The **covariance matrix** attached to the random vector $X = \big(f(x_1), f(x_2), \dots, f(x_n)\big)$ of a sample plan is given by

$$\Sigma_{XX} = \begin{pmatrix} \sigma_{11} & \cdots & \sigma_{1n} \\ \vdots & \ddots & \vdots \\ \sigma_{n1} & \cdots & \sigma_{nn} \end{pmatrix}$$

where the coefficients are defined as the covariance

$$\sigma_{ij} = \mathbb{E}\Big(f(x_i) f\big(x_j\big)\Big) - \mu_i \mu_j$$

of the random variables $f(x_i)$ and $f\big(x_j\big)$ viewed as random variables and $\mu_i = \mathbb{E}\big(f(x_i)\big)$ and $\mu_j = \mathbb{E}\big(f(x_i)\big)$ are the respective means. Now we pick an input vector $x_0 \in \mathbb{R}^k \backslash \{x_1, \dots, x_n\}$ with random variable $Y = f(x_0): \Omega \to \mathbb{R}$ and mean $\mu_Y = \mathbb{E}(Y) = \mathbb{E}\big(f(x_0)\big)$ and observe the joint random vector $(Y, X) = (f(x_0), f(x_1) \dots, f(x_n))$ whose covariance matrix is given by

$$\Sigma = \begin{pmatrix} \Sigma_{YY} & \Sigma_{YX} \\ \Sigma_{XY} & \Sigma_{XX} \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}$$

where $\Sigma_{YX} = \big(\sigma_{YX_1}, \dots, \sigma_{YX_n}\big)$ is a lying vector with components

$$\sigma_{YX_i} = \mathbb{E}\big(f(x_0) f(x_i)\big) - \mu_0\, \mu_i \quad \text{and} \quad \Sigma_{YY} = \mathbb{E}(f(x_0) f(x_0) - \mu_0 \mu_0.$$

By definition one easily calculates that $\Sigma_{YX} = \Sigma_{XY}$.

2.3 Statistical Prediction

Given that the values $f(x_1), \dots, f(x_n)$ are calculated by the computer simulated program and let $x_0 \in \mathbb{R}^k \backslash \{x_1, \dots, x_n\}$ be a new input vector that has not been calculated yet. We want to **guess** the values $f(x_0)$ without running it through the computer simulated calculation. Since we interpret the computer simulated function as an Gaussian process $f: \mathbb{R}^k \times \Omega \to \mathbb{R}$ and the outcomes as a random variables $X_1 = f(x_1), \dots, X_n = f(x_n)$ we may predict or guess $f(x_0)$. But first we need to define what a good guess or an optimal prediction is. In the framework of Gaussian processes this is done as follows: A function $p: \mathbb{R}^k \to \mathbb{R}$ is called **prediction function** of $Y = f(x_0)$ given $X_1 = f(x_1), \dots, X_n = f(x_n)$ if for all functions $g: \mathbb{R}^k \to \mathbb{R}$ the inequality

$$\mathbb{E}\left[\big(Y - p(X)\big)^2\right] \leq \mathbb{E}\left[\big(Y - g(X)\big)^2\right]$$

holds. Here $p(X) = p(X_1, \dots, X_n)$ and $g(X) = g(X_1, \dots, X_n)$ are random variables defined by $p(X) = p \circ X$ and $g(X) = g \circ X$. It is a fundamental mathematical theorem that the **prediction function** $p: \mathbb{R}^k \to \mathbb{R}$ of $Y = f(x_0)$ given $X_1 = f(x_1), \dots, X_n = f(x_n)$ is uniquely determined by the formula

$$p(x) = \mu_Y + \Sigma_{YX}\Sigma_{XX}^{-1}(x - \mu_X)$$

being the core object of your research project. Note that the lying vector
Here Σ_{XX}^{-1} is the inverse matrix of the covariance matrix of $X = (X_1, \dots, X_n)$ and Σ_{YX} is the lying vector described above. Note that a considerable time is taken in to account by the computer in order to calculate the inverse Σ_{XX}^{-1}.

2.4 A model for the prediction function

Given a sample plan $\{x_1, \dots, x_n\} \subset \mathbb{R}^k$ and having calculated only the outputs

$$y_1 = f(x_1),\ y_2 = f(x_2), \dots,\ y_n = f(x_n)$$

of the sample plan (still not knowing the whole of f) we want to make assumption on the mean μ_Y and the mean vector oft he inputs $\mu_X = \left(\mu_{X_1}, \dots, \mu_{X_n}\right) \in \mathbb{R}^n$. Another assumption we make is on the coefficients σ_{ij} of the covariance matrix Σ_{XX}. We choose these coefficients to be so called **radial basis functions (RBF)** defined by

$$\sigma_{ij} = \exp\left(-\sum_{k=1}^{n} \theta_k \left|x_{ik} - x_{jk}\right|^2\right)$$

where the vector $\theta = (\theta_1, \dots, \theta_n) \in \mathbb{R}^n$ conists of co called **hyperparameters**. We now turn our interest to the mean. We assume the Gaussian process to be **stationary**, i.e. that the mean function $\mu(x) = \mathbb{E}\big(f(x)\big) = \mu$ is constant for all $x \in \mathbb{R}^k$. After that assumption we define **estimated mean** by

$$\tilde{\mu} = \frac{\mathbf{1}^t \Sigma_{\mathrm{XX}}^{-1} y}{\mathbf{1}^t \Sigma_{\mathrm{XX}}^{-1} \mathbf{1}}$$

where $y = \big(f(x_1), f(x_2), \dots, f(x_n)\big) \in \mathbb{R}^n$ and $\mathbf{1} = (1,1, \dots, 1) \in \mathbb{R}^n$. The estimated mean is obtained by making a linear ansatz and minimizing the mean squared error. This method is also known as **Kriging the mean**. With these choices maid we get the optimal predictor

$$p(x) = \tilde{\mu} + \Sigma_{YX}\Sigma_{XX}^{-1}(x - \tilde{\mu})$$

which depends on the sample plan $\{x_1, \dots, x_n\} \subset \mathbb{R}^k$ and the hyperparameters $\theta = (\theta_1, \dots, \theta_n) \in \mathbb{R}^n$. Note that the components of the lying vector $\Sigma_{YX}(x) = \left(\sigma_{YX_1}, \dots, \sigma_{YX_n}\right)$ depends on $x \in \mathbb{R}^k$. The exact formula is given further below.

$$\sigma_{YX_i}(x) = \exp\left(-\sum_{k=1}^{n} \theta_k |x - x_{ik}|^2\right)$$

We have modelled the optimal predictor which will approximated the energy balance

$$p \approx f$$

where the prediction function p approximates the energy function f in a satisfactory way if the sample plan $\{x_1, \dots, x_n\} \subset \mathbb{R}^k$ has good **space filling properties**, whose meaning will be explained in the next section. Finding the **optimal hyperparameters** $\theta = (\theta_1, \dots, \theta_n) \in \mathbb{R}^n$ is also a topic that will be of concern.

2.5 Construction of the best sample plan as the best latin hypercube

Note that the computer simulated energy balance $f: \mathbb{R}^k \to \mathbb{R}$ will not be defined on the whole of the Euclidean space $\mathbb{R}^k$. Most components $x_{ij} \in \mathbb{R}$ in the input vector $x_i = (x_{i1}, \dots, x_{ik}) \in \mathbb{R}^k$ will be positive or have lower and upper bounds, i.e. $x_{ij} \in [a_i, b_i]$. This means that the computer simulated energy balance f is defined on a **design domain** defined by the hypercube

$$D \coloneqq [a_1, b_1] \times \cdots \times [a_k, b_k] \subset \mathbb{R}^k.$$

Note that for any building in planning the design domain has to be determined separately by the civil engineers. According to local conditions, cost calculation and owner desires the design domain may be choose differently in any situation.
We want to choose a sample plan $X \coloneqq \{x_1, \dots, x_n\} \subset D$ that covers the design domain D best. We choose the sample plans $X \coloneqq \{x_1, \dots, x_n\} \subset D$ to be in the category of *latin hypercubes*. In order to explain the notion of latin hypercube we divide any factor $I_i = [a_i, b_i]$ in subintervals

$$I_i = I_{i1} \cup \cdots \cup I_{in}.$$

Any permutation $\sigma: \{1, \dots, n\} \to \{1, \dots, n\}$ determines subcubes in D denoted by

$$Q_\sigma = I_{1\sigma(1)} \times \cdots \times I_{k\sigma(n)}$$

that cover the design domain, i.e. $D = \bigcup_\sigma Q_\sigma$ where the union runs over all permutations σ.
A sample plan $X = \{x_1, \dots, x_n\} \subset D$ is called **latin hypercube** if the following is satisfied

i. Any input vecor of the sample plan $X = \{x_1, \dots, x_n\}$ lies in the interior of some subcube Q_σ.
ii. The components of any pair of different input vectors $x_i \neq x_j$ must lie in different subcubes.

In the next picture one can see a two-dimensional example of a latin hypercube. The sample plan $X = \{x_1, \dots, x_{10}\}$ consists of ten two-dimensional vectors represented by bullets. The condition (i) says that any bullet must lie inside a square. Condition (ii) says that the projections of any to bullets to the x-axis or y-axis must be different:

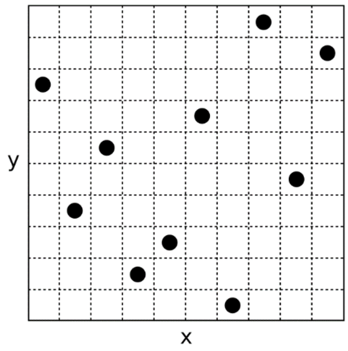


Figure 3: Two-dimensional Latin hypercube with 10 sample points. As one can see any input vector lies in the interior some subcube satisfying (i). Any two input vectors have different vertical and horizontal components.

As you can see in Figure 3 there are some regions in the design domain D that are not sampled well. We would like to *"measure"* how well a sample plan fills the space. To do so we will work with the Euclidean distance

$$||x_i - x_j|| = \sqrt{\sum_{\ell=1}^{k} |x_{i\ell} - x_{j\ell}|^2}$$

where $x_i = (x_{i1}, \ldots, x_{ik}) \in \mathrm{D}$ and $x_j = \left(x_{j1}, \ldots, x_{jk}\right) \in D$. The set of distances occurring in the sample plan $X = \{x_1, \ldots, x_n\}$ denoted by $\{||x_i - x_j|| : x_i, x_j \in X\}$ is written as an ordered set

$$d_1 < d_2 < \cdots < d_m$$

are called the **distances occurring in** X with **frequentnesses**

$$n_1 < n_2 < \cdots < n_m.$$

Note that $n_i(X)$ is the number of distances $d_i(X)$ occuring in the sample plan $X = \{x_1, \ldots, x_n\}$ measured in the Euclidean norm. Now there is a function due to Morris and Mitchell [Reference], which measures the space filling property of a sample plan X. Let $q \in \mathbb{N}$ be a natural integer. The **Moris-Mitchell function** is defined on sample plans $X = \{x_1, \ldots, x_n\} \subset D$ via

$$X \mapsto \Phi_q(X) = \left(\sum_{\ell=1}^{m} n_\ell \, d_\ell^{-q}\right)^{\frac{1}{q}}$$

The smaller the value $\Phi_q(X)$ attached to some sample plan X is, the better the space filling property of X is. This means that we must find sample plans that are latin hypercubes and that minimize the Moris-Mitchel function $X \mapsto \Phi_q(X)$. The third and last space fillingnes property is measured by the minimax-principle. We start with the initial set of $N \in \mathbb{N}$ different sample plans

$$S_0 = \{X_1, \dots, X_N\}$$

such that any sample plan consists of n input vectors $X_i = \{x_1, \dots, x_n\} \subset D$. We define first subsets of the initial collection by

$$S_1^{max} = \{X \in S_0 |\ d_1(X)\ ist\ maximal\}$$
$$S_1^{mima} = \{X \in S_1^{max} |\ n_1(X)\ is\ minimal\}.$$

Note that the sampling plans in S_1^{mima}maximize d_1 and among the plans for which this is true, minimize n_1. We recursively define

$$S_\ell^{max} = \{X \in S_{\ell-1}^{mima} |\ d_\ell(X)\ ist\ maximal\}$$
$$S_\ell^{mima} = \{X \in S_\ell^{max} |\ n_\ell(X)\ is\ minimal\}.$$

This construction yields a filtration of collections of sampling plans

$$S_0 \supset S_1^{mima} \supset S_2^{mima} \supset \dots \supset S_m^{mima}$$

Any sample plan $X_* \in S_m^{mima}$ is called **minimax** of $S_0 = \{X_1, \dots, X_n\}$ also written $X_* \in S_0$.

2.6 Evolutionary operation

We proceed by an evolutionary method used in biology: We pick a latin hypercube (called parent) and perturb it a P number of times yielding a population (called offsprings). Among this population of mutations, we pick the latin hypercube that minimizes the Moris-Mitchell function $\Phi_q(X)$ for some fixed $q \in \mathbb{N}$. We start now with the above constructed minimal latin hypercube and iterate this procedure N times that yields a minimum over all N generations denoted by X^q.

(1) Start with a latin hypercube $X_1 \subset D$ and change this latin hypercube randomly in order to get a **population** $S_1^{Pop} = \{X_1^1, X_1^2, \dots, X_1^P\}$ of latin hypercubes and pick the one $X_1^{pop} \in S_1^{Pop}$ among this population with the minimal value $\Phi_q(X)$. The parameter $P \in \mathbb{N}$ is called the **population number** and is the number of mutated individuals in the evolutionary operation

(2) We start with $X_1^{pop} \in S_1^{Pop}$ of Step (1) and create a new population of latin hypercubes S_2^{Pop} as in Step (1) where we pick the sample plan $X_2^{pop} \in S_2^{Pop}$ with minimal value $\Phi_q(X)$ among this population. We **iterate** this procedure $N \in \mathbb{N}$ times to get a finite set of latin hypercubes $\{X_1^{pop}, \dots, X_N^{pop}\}$. The last latin hypercube $X^q = X_N^{pop} \in S_N^{Pop}$ that occurred by minimizing $\Phi_q(X)$ among all S_N^{Pop} populations.

(3) We perform Step (1) and Step (2) for different values $q_1, q_2, \dots, q_M$ which yields a collection of sample plans $S_0 = \{X^{q_1}, X^{q_2}, \dots, X^{q_M}\}$ within the category of latin hypercubes.

(4) We choose a **minimax** $X_* \in S_0$ which is called **best latin hypercube.**

The best latin hypercube palys an important role this research project. The prediction function

$$p: D \subset \mathbb{R}^k \to \mathbb{R}$$

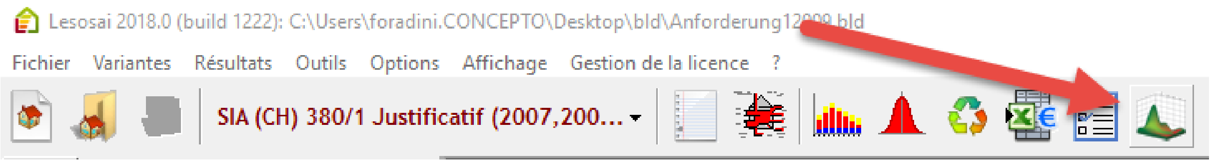
depends on the sample plan $X = \{x_1, \ldots, x_n\} \subset D$ which is from now on choose to be the **best latin hypercube** constructed in the procedure defined by Step (1) to Step (4).

3 Implementation of the prediction function in Lesosai

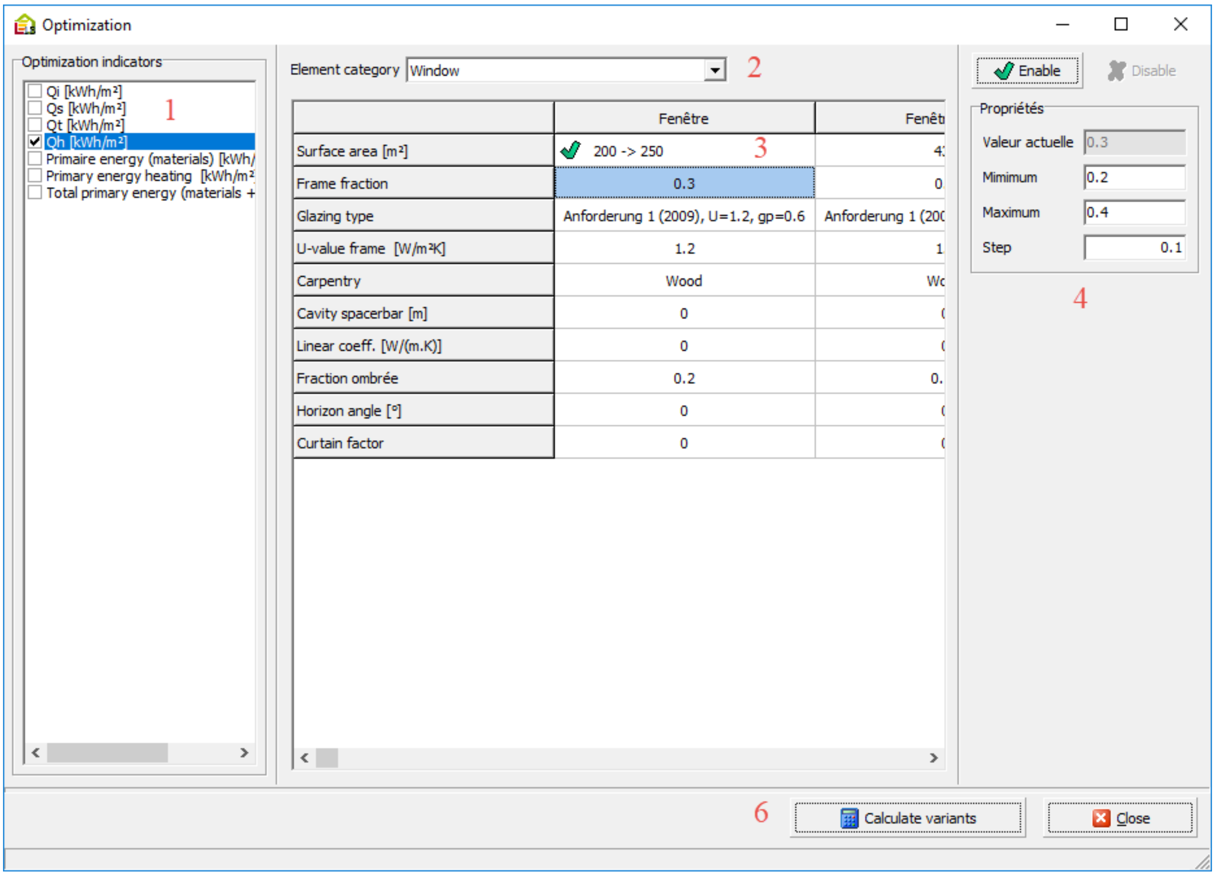
3.1 Implementation first version for the customers

We implemented in Lesosai the possibility to do a multi-calculation as first step of the project. This first implementation was based did not use the mathematical algorithms studied in this project. This step was needed to have a Lesosai adapted to start the calculations defining limits and steps on value. This first implementation is usable only with the SIA380/1 that is a monthly calculation but not usable with hourly calculations as defined in SIA2044.

Start screen:



Calculations options:

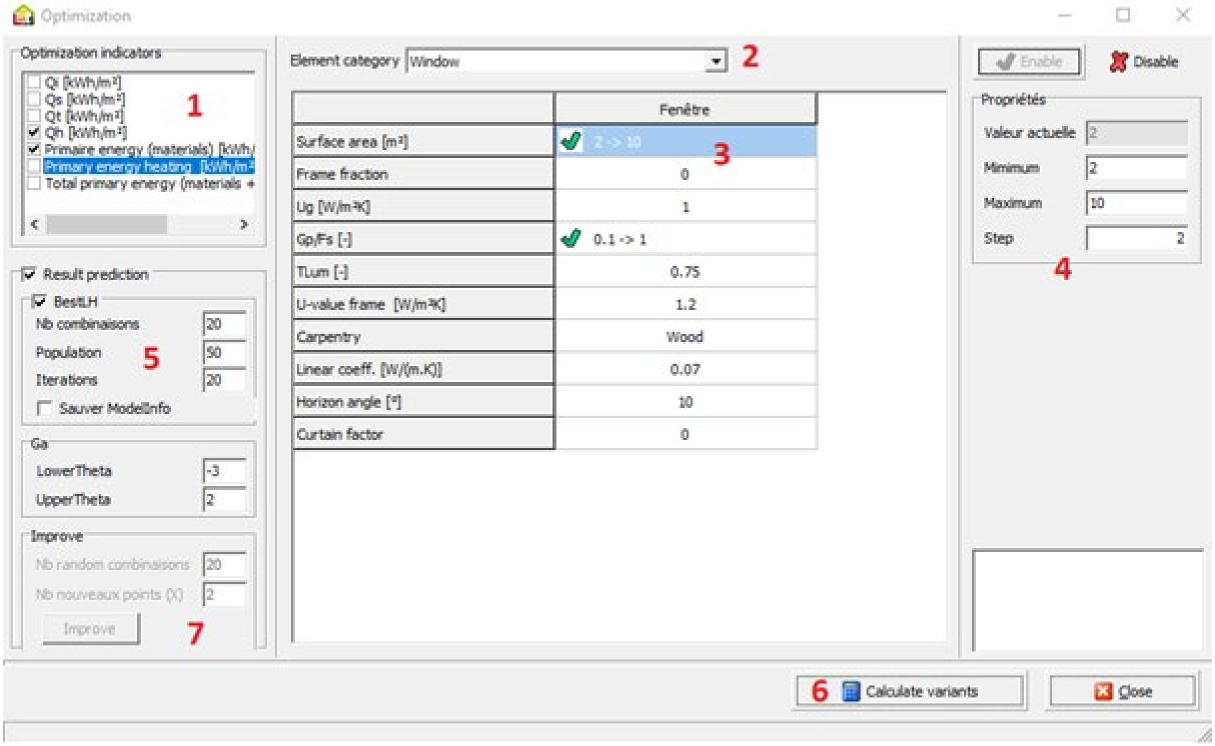


1. Choice of comparison indicators (results for various thermal & environmental calculations, depending on the standard selected in Lesosai). Currently, it is only possible to perform predictions using a single indicator.
2. List of building elements (walls, floors, windows, etc.)
3. Properties of currently selected element (2)
4. When selecting a property (3), the user can enable it. When enabled, a property becomes a variable.
5. The minimum and maximum values of this property as well as the incremental steps must be defined.
6. Start the algorithm

This first implementation is used since more than 2 years from the customers from Lesosai.

3.2 Implementation of the algorithms for the project

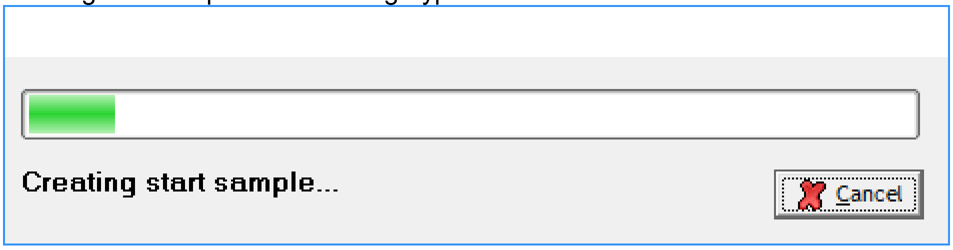
The second step in the project was to introduce the previsions algorithms for the project.



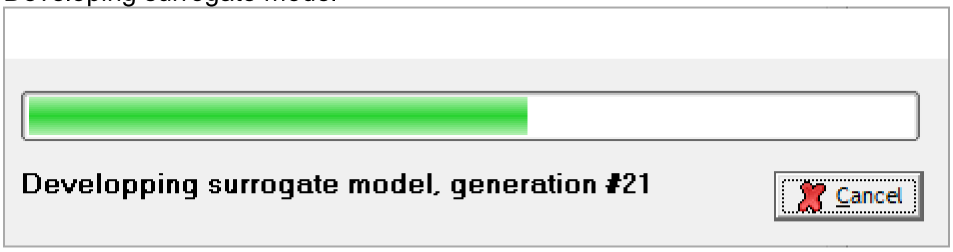
This implementation allowed to test the methodology and gives the possibilities to check the influence of the different parameters.
Changed points from the first implementation:

4. *When selecting a property (3), the user can enable it. When enabled, a property becomes a variable. The minimum and maximum values of this property as well as the incremental steps must be defined.* They are not used for prediction but we will use them to know which combinations of parameters we will try to predict results for once the surrogate model has been developed. There is no limit on the number of properties that can become variable. In the tests we made, we worked with a low number of variables (two or three). In the above example, the windows area and Gp values as well as the external walls U-value are set up as variable.
5. When checking the "results prediction" checkbox, some starting parameters have to be defined in order to build the hypercube and setup the prediction algorithm. When this checkbox is not checked, each possible combination of parameter is calculated, no predictions are made. Both modes can be used successively in order to get prediction and calculated results for the same combinations of parameters and evaluate the quality of the predictions.
6. Start the algorithm
7. Improve predictions if results are not good enough by adding new simulated combinations

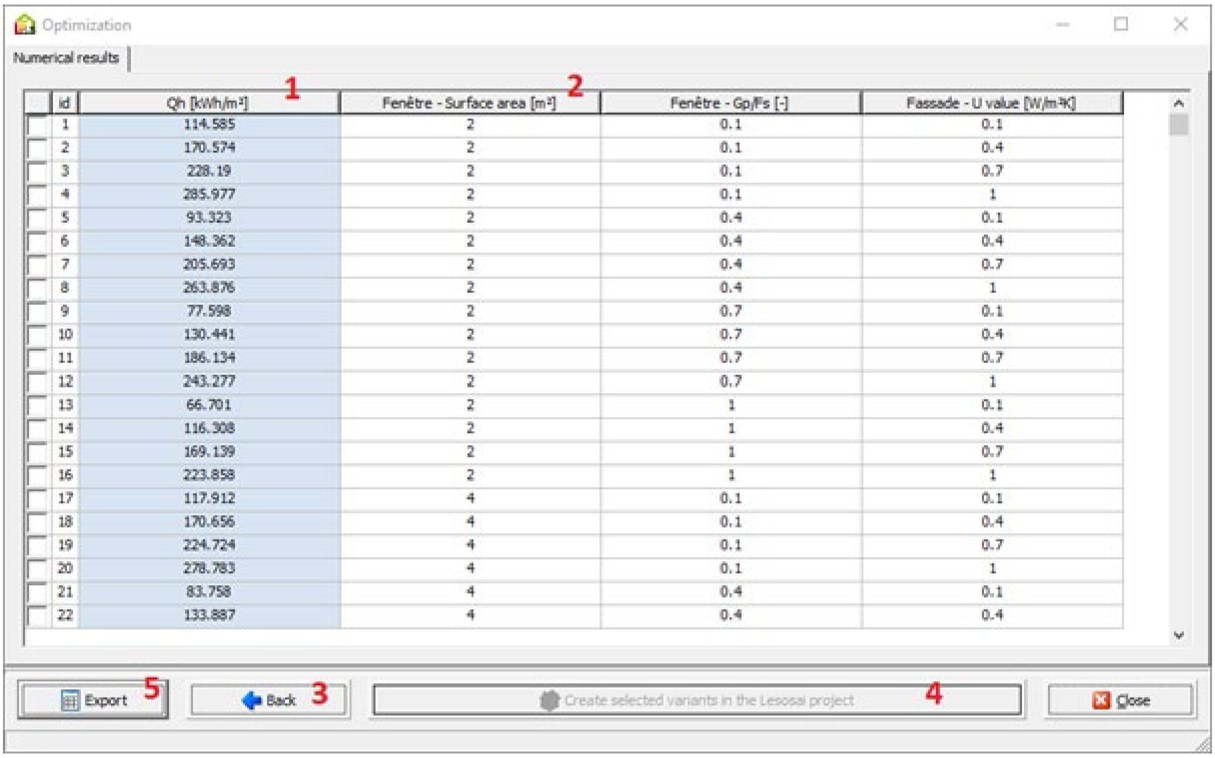
Creating the best possible starting hypercube



Developing surrogate model



Results



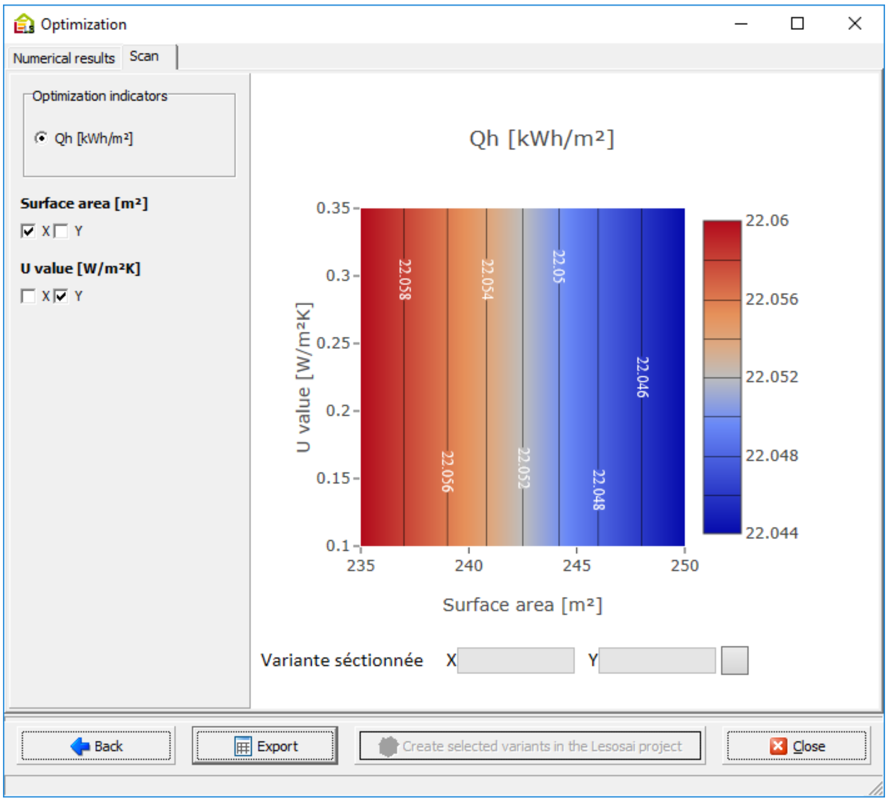
id	Qh [kWh/m²]	Fenêtre - Surface area [m²]	Fenêtre - Gp/Fs [-]	Fassade - U value [W/m²K]
1	114.585	2	0.1	0.1
2	170.574	2	0.1	0.4
3	228.19	2	0.1	0.7
4	285.977	2	0.1	1
5	93.323	2	0.4	0.1
6	148.362	2	0.4	0.4
7	205.693	2	0.4	0.7
8	263.876	2	0.4	1
9	77.598	2	0.7	0.1
10	130.441	2	0.7	0.4
11	186.134	2	0.7	0.7
12	243.277	2	0.7	1
13	66.701	2	1	0.1
14	116.308	2	1	0.4
15	169.139	2	1	0.7
16	223.858	2	1	1
17	117.912	4	0.1	0.1
18	170.656	4	0.1	0.4
19	224.724	4	0.1	0.7
20	278.783	4	0.1	1
21	83.758	4	0.4	0.1
22	133.887	4	0.4	0.4

1. Column in blue: predicted results for the indicator chosen
2. Columns in white: variable parameters. Each row of the table corresponds to a combination of parameters
3. Go back to previous screen in order to try to improve the prediction by performing new simulations and adding some new points
4. Create a new Lesosai variant which will take the same parameters as the selected row in the above table
5. Export results to Excel.

This version is not available to customers.

3.3 Final implementation for the customers

In 2019 we will make the step 3.2 available to the customers, we are improving the ergonomics giving, for example, a graphically possibility of analysis:



4 Testing the influence parameters of the best latin hypercube *bestln*

There are two possible approaches to work out whether parameter have an influence on a function or not. One way is to study the properties of the function, the other is to calculate multiple variants of function values of the function with different choices of parameters in order to compare the values with respect to the change of parameters.
Since we have not too much control on the algorithm *bestln* we chose to just vary the possible parameters, construct the best latin hypercube and generate the prediction function based on the best latin hypercube.

4.1 Experimental approach

The construction of the best latin hypercube *bestln* and the prediction function p_{Del} are both implemented in *Delphi* within the framework of *Lesosai.* Hence, we have done all calculation within the software *Lesosai*. Before the calculation can start a building model (as a study case) has to be created with respect to which one can perform the calculation. *E4Tech* already had such a building in planning to test the dependence of the implementation *bestln*. In this study case *E4Tech* suspects that the prediction function $p_{Del}: \mathbb{R}^3 \to \mathbb{R}$ has a big deviation from the energy balance $f_{Del}: \mathbb{R}^3 \to \mathbb{R}$. Since p_{Del} is a bad approximation of f_{Del} it is expected that changes in the bad setting have a bigger influence than changes of the parameters in a sufficiently good setting. As next we explain the components of the input vector $x = (x_1, x_2, x_3) \in \mathbb{R}^3$. Any input vector consists of the building parameters

$x_1 =$ *(total area of windows)* contained in $[2,10]$

$x_2 =$ *(window g_p value)* contained in $[2,10]$

$x_3 =$ *(wall U value)* contained in $[0.1,0.7]$.

The design domain of this study case is then determined by the three-dimensional hypercube

$$D = [2,10] \times [0.1,0.7] \times [0.1,0.7] \subset \mathbb{R}^3.$$

Now note that we will take a partition of $[2,10]$ of step length 1 and partitions of the two intervals $[0.1,0.7]$ of step length 0.1 respectively. This leads to 441 different input vectors are called **investigation plan** and is denoted by

$$G = \{x_1, x_2, \dots, x_{441}\} \subset D.$$

The investigation plan is a grid distributed over the cube $D = [2,10] \times [0.1,0.7] \times [0.1,0.7] \subset \mathbb{R}^3$. We will test now the function on every input vector $x_i \in \{x_1, x_2, \dots, x_{441}\}$ which yields 441 values for

$$p_{Del}(x_i) \quad \text{and} \quad f_{Del}(x_i) \quad \text{respectively.}$$

These 441 input vectors for each function are investigated. A big part in the understanding of the big picture is the fact that p_{Del} depends on the best latin hypercube *bestln* used to define it. We want to change the parameters used to define *bestln* and see for which parameters p_{Del} becomes the best prediction of f_{Del}. In order to measure the differences $p_{Del}(x_i)$ and $f_{Del}(x_i)$ we will introduce a bunch of statistical quantities. The *absolute difference function* and the *relative difference function* defined via

$$abs(x_i) = p_{Del}(x_i) - f_{Del}(x_i) \qquad \text{and} \qquad rel(x_i) = \frac{p_{Del}(x_i) - f_{Del}(x_i)}{f_{Del}(x_i)}$$

The **mean difference** of the hole investigation plan $G = \{x_1, x_2, \dots, x_{441}\}$ is defined via

$$\overline{m}_G = \frac{1}{441}\sum_{i=1}^{441} rel(x_i)$$

The **standard deviation** of the whole investigation plan $G = \{x_1, x_2, \dots, x_{441}\}$ is defined by

$$s_G = \sqrt{\frac{\sum_{i=1}^{441}(rel(x_i) - \overline{m})^2}{441}}$$

We also consider the interval $[\overline{m} - s_G, \overline{m} + s_G]$. Note that the chance that a measured quantity lies in the interval $[\overline{m} - s_G, \overline{m} + s_G]$ is 66%.

4.2 Influence parameters

Since the best latin hypercube *bestln* is constructed by the evolutionary process described above we have the following parameters that can be choosen.

$P =$ *(Population)*

$N =$ *(Iteration)*

$n =$ *(combination)*

where the combination n is the number of input vectors in any latin hypercube.
Since *bestln* depends on the triple (n, P, N) so does the prediction function p_{Del}. That is why we denote it by

$$p_{Del}(n, P, N): \mathbb{R}^3 \to \mathbb{R}$$

yielding for any triple (n, P, N) a prediction function $p_{Del}(n, P, N)$. We want to find the best triple such that the prediction function approximates

$$p_{Del}(n, P, N) \approx f_{Del}$$

the energy balacance best. By theoretical consideration all three parameters have a monotone rising influence. The parameter combination n has the biggest influence, since the more input vectors we have in the design domain D the exacter $p_{Del}(n, P, N)$ is going to approximate f_{Del}. The theory shows that the prediction function **converges** towards the energy balance

$$p_{Del}(n, P, N) \to f_{Del} \quad \text{for} \quad n \to \infty$$

However, increasing the combination n will not necessarly lead to a better latin hypercube if the population P and the iteration N because the spacefilling property of the best latin hypercube would not increase sufficiently. As it is not possible to get lower with our choices, convergence is the only option. There is a major flaw in making the combination n arbitrarily big: It will take a long time to generate the best latin hypercube *bestln* ! The increase of the population P and N make the prediction function also better but have an upper limit, where the time cost becomes too expensive.

4.3 Testing tables

We investigate the prediction function in **dependence on the combination**. We plug in the values

$n = 10$	$n = 20$	$n = 30$	$n = 40$	$n = 50$	$n = 60$	$n = 70$

where the population $P = 50$ and the iteration $N = 20$ are fixed.

(**n**,P,N)	(**10**,50,20)	(**20**,50,20)	(**30**,50,20)	(**40**,50,20)	(**50**,50,20)	(**60**,50,20)	(**70**,50,20)
$\overline{m}_G$	0.434%	0.074%	-0.004%	0.026%	0.035%	0.072%	0.149%
s_G	4.10%	0.93%	0.32%	0.23%	0.18%	0.53%	0.57%
$\overline{m}_G + s_G$	4.536%	1.002%	0.313%	0.258%	0.212%	0.603%	0.719%
$\overline{m}_G - s_G$	-3.669%	-0.854%	-0.320%	-0.205%	-0.143%	-0.459%	-0.422%
$\max(rel(x_i))$	22.066%	5.500%	1.582%	1.219%	1.280%	3.551%	3.935%
$\min(rel(x_i))$	-10.587%	-2.937%	-1.901%	-0.768%	-0.653%	-2.327%	-2.661%

Table 4: Relative difference by changing combination

We see that if the number of combinations is chosen to be $n = 30$ we get the best prediction possible. As next we investigate the prediction function in **dependence on the population**. We plug in the values

$P = 10$	$P = 30$	$P = 50$	$P = 70$	$P = 90$	$P = 110$	$P = 130$

where the population $n = 20$ and the iteration $N = 20$ are fixed.

(n,**P**,N)	(20,**10**,20)	(20,**30**,20)	(20,**50**,20)	(20,**70**,20)	(20,**90**,20)	(20,**110**,20)	(20,**130**,20)
$\overline{m}_G$	-0.133%	-0.813%	0.074%	-0.014%	-0.025%	0.005%	-0.017%
s_G	0.93%	2.64%	0.93%	1.11%	1.13%	0.88%	1.11%
$\overline{m}_G + s_G$	0.797%	1.829%	1.002%	1.094%	1.105%	0.881%	1.094%
$\overline{m}_G - s_G$	-1.064%	-3.454%	-0.854%	-1.122%	-1.154%	-0.871%	-1.129%
$\max(rel(x_i))$	4.782%	5.813%	5.500%	9.893%	5.977%	5.921%	4.214%
$\min(rel(x_i))$	-3.654%	-16.121%	-2.937%	-3.648%	-5.149%	-3.860%	-5.789%

Table 5: Relative difference by changing population

We see that if the number of the population is chosen to be $P = 110$ we get the best prediction possible. However, there is no strong or significant dependence on the population P.
As next we investigate the prediction function in **dependence on the iteration**. We plug in the values

$N = 10$	$N = 30$	$N = 50$	$N = 70$	$N = 90$	$N = 110$	$N = 130$

where the population $n = 20$ and the iteration $P = 50$ are fixed.

(n,P,**N**)	(20,50,**10**)	(20,50,**20**)	(20,50,**30**)	(20,50,**40**)	(20,50,**50**)	(20,50,**60**)	(20,50,**70**)
$\overline{m}_G$	-0.103%	0.074%	0.230%	-0.219%	-0.284%	-0.053%	0.136%
s_G	1.11%	0.93%	0.94%	0.97%	1.31%	0.90%	1.24%
$\overline{m}_G + s_G$	1.009%	1.002%	1.170%	0.747%	1.023%	0.852%	1.371%
$\overline{m}_G - s_G$	-1.214%	-0.854%	-0.710%	-1.185%	-1.590%	-0.958%	-1.099%
$\max(rel(x_i))$	2.907%	5.500%	9.240%	4.300%	7.332%	4.741%	10.937%
$\min(rel(x_i))$	-4.959%	-2.937%	-2.249%	-4.845%	-6.595%	-2.795%	-4.357%

Table 6: Relative difference by changing iteration

We see that if the number of the population is chosen to be $N = 60$ we get the best prediction possible. However, there is no strong or significant dependence on the iteration N.

4.4 Discussion

The results of the theoretical approach and those from the experimental observation do differ which does not mean that our method is wrong. By the theoretical consideration all three parameters (n, P, N) should make the prediction function $p_{Del}(n, P, N)$ converge towards f_{Del} in case we choose the parameters bigger and bigger. We see that there are parameter upper bounds where the prediction function does not necessarily increase.
In order to find a good latin hypercube in a suitable time all three parameters have to be coordinated. However, there is no formula that can give a combination of the three parameters to find a good latin hypercube.
By the experimental approach no clear influence was found especially not a monotone rising influence.

As a conclusion of this study there is no statement at all about the influence. We propose the following choice that is also proposed in [3]:

> "If the number of parameters $k < 20$ is smaller than twenty-one should pick the number of combinations to be $n < 500$, otherwise pick the number of combinations to be bigger than 500."

This completes the discussion.

5 Testing the implementation of the prediction function

We have taken codes from [3] where the prediction function p_{Mat} is implemented in Mathlab and have written codes in *Delphi* that give the prediction function p_{Del} within the framework of *Lesosai*. This means that we are dealing with two computer simulated functions

$$p_{Mat} : \mathbb{R}^k \to \mathbb{R} \qquad \text{and} \qquad p_{Del} : \mathbb{R}^k \to \mathbb{R}$$

that are coded in *Mathlab* and *Delphi* respectively. It is not easy to compare two computer simulated functions written in two different codes. Knowing that the prediction function in *Mathlab* approximates the time expensive energy balance f_{Mat} and showing additionally that

$$p_{Mat} = p_{Del},$$

we have a prediction function p_{Del} that approximates f_{Del}. It is not possible to translate the two codes like a usual language therefore it is necessary to compare the results. This chapter is not about to check whether the prediction function is a good algorithm or not, it is about checking whether the implementation into *Delphi* code has been done correctly. Since *E4Tech* was not in possession of a *Mathlab* license the data had to be transferred via Email between the two parties (*E4Tech* and *BFH*) being very time consuming. If an error would occur, we had to check whether this error was because of a bug in the code or a miscommunication issue.

5.1 Method of Testing

For the calculation of the values by *Matlab* and by *Delphi* the same *latin hypercube* must be used used since the prediction function depends of the latin hypercube used to define it. The values of three points were calculated and compared. Which points to choose is not relevant even if the results were different by choosing different points. However, they should not differ between the two calculation methods. The calculated values were copy pasted into the Microsoft program Excel for statistical evaluation. All displayed values were rounded to the fourth decimal. For the statistical evaluation the exact values were used in order not to lose precision. The statistical evaluation is done via absolute and relative difference. This we explain in the following. We want to measure how much the functions differ from each other. This is measured by the *absolute difference function* and the *relative difference function* defined via

$$abs(x) = p_{Mat}(x) - p_{Del}(x) \qquad \text{and} \qquad rel(x) = \frac{p_{Mat}(x) - p_{Del}(x)}{p_{Mat}(x)}$$

The relative difference was chosen to compare the different values in order to find possible weaknesses or bugs in the implementation of p_{Del} with no influence of the original value size.

5.2 Testing the prediction function and the results

As mentioned in the previous chapter the prediction function has the following structure

$$p(x) = \tilde{\mu} + \Sigma_{YX}\Sigma_{XX}^{-1}(x - \tilde{\mu})$$

that is implemented in two different codes. Note that the lying vector $\Sigma_{YX} = (\sigma_{YX_1}, \dots, \sigma_{YX_n})$ has components that depend on $x \in \mathbb{R}^k$ through the equation

$$\sigma_{YX_i}(x) = \exp\left(-\sum_{k=1}^{n} \theta_k |x - x_{ik}|^2\right).$$

The covariance matrix and the estimated mean of the *Matlab* code are denoted by

$$\Sigma_{YX}^{Mat} \in \mathbb{R}^n \qquad \text{and} \qquad \tilde{\mu}_{Mat} \in \mathbb{R}$$

whereas the the covariance matrix and the estimated mean implemented in the Delphi code are denoted by

$$\Sigma_{YX}^{Del} \in \mathbb{R}^n \qquad \text{and} \qquad \tilde{\mu}_{Del} \in \mathbb{R}.$$

Since the covariance vector and the estimated mean are used to define the prediction function, we check whether they are well-implemented. We check whether the equalities

$$\Sigma_{YX}^{Mat}(x) = \Sigma_{YX}^{Del}(x) \qquad \text{and} \qquad \tilde{\mu}_{Mat} = \tilde{\mu}_{Del}$$

hold. This will be done by calculating the prediction function, the estimated mean and the covariance vector at different points. We will display three tables in the following that explore the correctness of the implementation.

We have done a lot of calculations and comparisons and have not been able to detect an error in the implementation. Any difference that has occurred in the numbers has bee

5.3 Testing tables

We pick the inputvector to be $x = (0,0,0) \in \mathbb{R}^3$ and evaluate the following objects

$p_{Mat}(0,0,0)$ $\tilde{\mu}_{Mat}$ $\Sigma_{YX}^{Mat}(0,0,0)$	$p_{Del}(0,0,0)$ $\tilde{\mu}_{Del}$ $\Sigma_{YX}^{Del}(0,0,0)$

and check whether they are identical.

Inputvector
$x = (0,0,0) \in \mathbb{R}^3$

f

Values of Matlab			Values of Delphi			Absolut difference $abs(x)$			Relative Difference $rel(x)$		
$p_{Mat}(x)$	$\tilde{\mu}_{Mat}$	$\Sigma_{YX}^{Mat}(x)$	$p_{Del}(x)$	$\tilde{\mu}_{Del}$	$\Sigma_{YX}^{Del}(x)$	p	$\tilde{\mu}$	$\Sigma_{YX}(x)$	p	$\tilde{\mu}$	$\Sigma_{YX}(0,0,0)$
76.1610	211.7191	0.9344	76.1610	211.7191	0.9344	0.0000	0.0000	0.0000	0.00%	0.00%	0.00%
		0.9836			0.9836			0.0000			0.00%
		0.9387			0.9387			0.0000			0.00%
		0.9411			0.9411			0.0000			0.00%
		0.9158			0.9158			0.0000			0.00%
		0.9835			0.9835			0.0000			0.00%
		0.8701			0.8701			0.0000			-0.01%
		0.9427			0.9427			0.0000			0.00%
		0.8860			0.8860			0.0000			0.00%
		0.8099			0.8099			0.0000			0.00%
		0.9272			0.9272			0.0000			0.00%
		0.8524			0.8524			0.0000			0.00%
		0.9461			0.9461			0.0000			0.00%
		0.9430			0.9430			0.0000			0.00%
		0.8789			0.8789			0.0000			0.00%
		0.8906			0.8906			0.0000			0.00%
		0.9985			0.9985			0.0000			0.00%
		0.8735			0.8735			0.0000			0.00%
		0.9774			0.9774			0.0000			0.00%
		0.9733			0.9733			0.0000			0.00%

Table 7: Comparison of the prediction function p, the estimated mean $\tilde{\mu}$ and the covariance vector Σ_{YX} at the inputvector $x = (0,0,0) \in \mathbb{R}^3$

We pick the inputvector to be $x = (0,0,0.33333333) \in \mathbb{R}^3$ and evaluate the following objects

$p_{Mat}(0,0,0.33333333)$	$p_{Del}(0,0,0.33333333)$
$\tilde{\mu}_{Mat}$	$\tilde{\mu}_{Del}$
$\Sigma_{YX}^{Mat}(0,0,0.33333333)$	$\Sigma_{YX}^{Del}(0,0,0.33333333)$

and check whether they are identical.

Inputvector
$x = (0,0,0.33333333) \in \mathbb{R}^3$

Values of Matlab			Values of Delphi			Absolut difference $abs(x)$			Relative Difference $rel(x)$		
$f_{Mat}(x)$	$\tilde{\mu}_{Mat}$	$\Sigma_{YX}^{Mat}(x)$	$f_{Mat}(x)$	$f_{Mat}(x)$	$\tilde{\mu}_{Mat}$	$\Sigma_{YX}^{Mat}(x)$	$f_{Mat}(x)$	$f_{Mat}(x)$	$\tilde{\mu}_{Mat}$	$\Sigma_{YX}^{Mat}(x)$	$f_{Mat}(x)$
131.9097	211.7191	0.9321	131.9097	211.7191	0.9321	0.0000	0.0000	0.0000	0.00%	0.00%	0.00%
		0.9956			0.9956			0.0000			0.00%
		0.9474			0.9474			0.0000			0.00%
		0.9721			0.9721			0.0000			0.00%
		0.9189			0.9189			0.0000			0.00%
		0.9840			0.9840			0.0000			0.00%
		0.8756			0.8756			0.0000			0.00%
		0.9653			0.9653			0.0000			0.00%
		0.9259			0.9259			0.0000			0.00%
		0.8439			0.8439			0.0000			0.00%
		0.9633			0.9633			0.0000			0.00%
		0.8933			0.8933			0.0000			0.00%
		0.9604			0.9604			0.0000			0.00%
		0.9712			0.9712			0.0000			0.00%
		0.9026			0.9026			0.0000			0.00%
		0.9226			0.9226			0.0000			0.00%
		0.9932			0.9932			0.0000			0.00%
		0.8918			0.8918			0.0000			0.00%
		0.9950			0.9950			0.0000			0.00%
		0.9654			0.9654			0.0000			0.00%

Table 8: Comparison of the prediction function p, the estimated mean $\tilde{\mu}$ and the covariance vector Σ_{YX} at the inputvector $x = (0,0,0.33333333) \in \mathbb{R}^3$

We pick the inputvector to be $x = (0,0,0.66666666) \in \mathbb{R}^3$ and evaluate the following objects

$p_{Mat}(0,0,0.66666666)$ $\tilde{\mu}_{Mat}$ $\Sigma_{YX}^{Mat}(0,0,0.66666666)$	$p_{Del}(0,0,0.66666666)$ $\tilde{\mu}_{Del}$ $\Sigma_{YX}^{Del}(0,0,0.66666666)$

and check whether they are identical.

Inputvector
$x = (0,0,0.66666666) \in \mathbb{R}^3$

Values of Matlab			Values of Delphi			Absolut difference $abs(x)$			Relative Difference $rel(x)$		
$f_{Mat}(x)$	$\tilde{\mu}_{Mat}$	$\Sigma_{YX}^{Mat}(x)$	$f_{Mat}(x)$	$f_{Mat}(x)$	$\tilde{\mu}_{Mat}$	$\Sigma_{YX}^{Mat}(x)$	$f_{Mat}(x)$	$f_{Mat}(x)$	$\tilde{\mu}_{Mat}$	$\Sigma_{YX}^{Mat}(x)$	$f_{Mat}(x)$
189.6673	211.7191	0.9121	189.6673	211.7191	0.9121	0.0000	0.0000	0.0000	0.00%	0.00%	0.00%
		0.9884			0.9884			0.0000			0.00%
		0.9378			0.9378			0.0000			0.00%
		0.9849			0.9849			0.0000			0.00%
		0.9044			0.9044			0.0000			0.00%
		0.9656			0.9656			0.0000			0.00%
		0.8643			0.8643			0.0000			0.00%
		0.9695			0.9695			0.0000			0.00%
		0.9490			0.9490			0.0000			0.00%
		0.8625			0.8625			0.0000			0.01%
		0.9816			0.9816			0.0000			0.00%
		0.9182			0.9182			0.0000			0.00%
		0.9563			0.9563			0.0000			0.00%
		0.9811			0.9811			0.0000			0.00%
		0.9091			0.9091			0.0000			0.00%
		0.9374			0.9374			0.0000			0.00%
		0.9691			0.9691			0.0000			0.00%
		0.8931			0.8931			0.0000			0.00%
		0.9935			0.9935			0.0000			0.00%
		0.9391			0.9391			0.0000			0.00%

Table 9: Comparison of the prediction function p*, the estimated mean* $\tilde{\mu}$ *and the covariance vector* Σ_{YX} *at the inputvector* $x = (0,0,0.66666666) \in \mathbb{R}^3$

5.4 Discussion

The probability that an error in the implementation has occurred is basically zero. As the results obtained from the different codes were transferred by copy and paste it is impossible to mistype. If a value would have been pasted in a wrong position the results would differ more than they do. The relative difference is not more than 0.01%. The difference can be explained by the different number of float digits that Matlab and Delphi calculate with. It is mentioned worthy that Delphi calculates in a higher number of float digits. Given the above explained examples and dozens of examples that we have not displayed we consider that the **implementation is correct.** There are no further investigations in this area.

6 References

[1] TEP Energy GmbH, Prognos AG, Basel, Infras AG, Bern, “Analyse des schweizerischen Energieverbrauchs 2000 - 2013 nach Verwendungszwecken”, 2014.

[2] SIA, Schweizerischer Ingenieur- und Architektenverein, “SIA 380/1 – Thermische Energie im Hochbau”, 2009.

[3] Alexander Forrester, Andras Sobester, Andy Keane, “Engineering Design via Surrogate Modelling: A Practical Guide”, Wiley, 2008.

[4] B. Ladevie *et al.*, “Analyses multicritères et méthode inverse en simulation énergétique du bâtiment”, 2012.

[5] SIA, Schweizerischer Ingenieur- und Architektenverein, "SIA 2044 - Klimatisierte Gebäude - Standard-Berechnungsverfahren für den Leistungs- und Energiebedarf", 2011.

7 Indices

7.1 Index of tables

7.2 Index of figures

A. Appendix

Optivitrage

Un nouvel outil pour améliorer l'efficacité des fenêtres en matière d'énergie, de confort et de dimensions

Flavio Foradini
Windays 2019 - Bienne

certifications & bilans écologiques et énergétiques de bâtiments — E4tech software

Questions

Combien de fenêtres faut-il mettre sur un bâtiment?

Quel type de fenêtre?

Ai-je besoin de climatiser?

certifications & bilans écologiques et énergétiques de bâtiments — E4tech software

Les effets des fenêtres

Le choix des fenêtres a un impact (augmentation des fenêtres signifie) :

- Sur les besoins énergétiques:
  - + de gains de chaud
  - + de besoins de froid
- Sur les installations techniques
- Sur le confort:
  - + d'éclairage naturel
  - + de risques de surchauffe

certifications & bilans écologiques et énergétiques de bâtiments — E4tech software

Méthode de travail classique

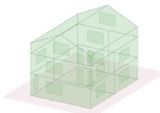
L'architecte fait le dessin et, avec le client, définit les objectifs.

Il l'envoie au physicien du bâtiment (expert) qui essaie rapidement d'obtenir des résultats.

Le projet repart vers l'architecte avec les informations de changements.

Le temps de travail est limité, donc la solution, bien qu'acceptable, n'est, souvent, pas idéale.

Le calcul du confort dans les cas limites est horaire.



4 certifications & bilans écologiques et énergétiques de bâtiments — E4tech software

Le projet **Optivitrage**

Un projet de l'Office Fédéral de l'Energie.

Participants:

- E4tech Software SA: logiciel Lesosai et connaissance des normes
- Haute école spécialisée bernoise - section Architecture: connaissance des fenêtres, de la physique du bâtiment et des méthodes mathématiques

Objectif de l'architecte:

- Avoir un bâtiment qui satisfait les lois suisses et le client

Objectif du projet :

- Minimiser le besoin d'énergie primaire d'utilisation en maintenant le confort
- tout en limitant le temps de calcul et de recherche de solution

5 certifications & bilans écologiques et énergétiques de bâtiments — E4tech software

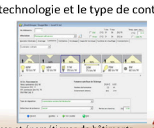
Méthode de travail - 1

L'architecte fait le projet de base et définit les objectifs avec le client.

Avec l'expert, l'architecte:

définit les marges de manœuvre (par exemple) sur :

- Les constructions : épaisseurs maximales des éléments, épaisseurs minimales de la structure porteuse, les couches de base, la position des matériaux
- Les fenêtres: dimensions minimales et maximales, valeurs U, Gp,...
- La ventilation: mécanique / naturelle
- L'éclairage: valeurs limite et cible, la technologie et le type de contrôle (par ex. manuel et automatique)
- ...



6 certifications & bilans écologiques et énergétiques de bâtiments — E4tech software

1

Méthode de travail - 2

L'expert lance un grand nombre de calculs qu'il analyse et, ensuite, propose plusieurs solutions au client. Elles peuvent avoir le même résultat.

Exemple énergétique:

Il est possible de mieux isoler les murs et de moins isoler les vitrages ou le contraire et avoir le même résultat.

Méthodologie

Le calcul horaire n'est pas un calcul simple, donc la méthodologie permet le développement d'un modèle succédané qui approche le modèle inconnu.

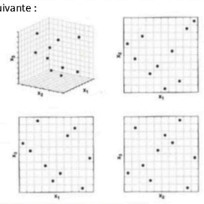
Etapes:

1. Préparation d'un échantillon de combinaisons de paramètres
2. Développement du modèle succédané (de substitution)
3. Exploration du modèle succédané pour trouver un ou plusieurs résultats
4. Amélioration éventuelle du modèle succédané

Méthodologie : échantillon

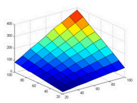
Le but de l'échantillon est qu'il remplisse l'espace.

La solution proposée est un échantillon de type « hypercube latin» aléatoire, qui est représenté par la figure suivante :



Méthodologie : modèle succédané

A partir de calculs «réels» sur l'échantillon, la méthode développe un modèle succédané qui permet de calculer rapidement tous les points voulus:

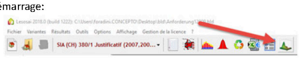


Des tests permettent d'évaluer la qualité du modèle. Si elle n'est pas suffisante, la méthode ajoute une autre série d'échantillons et crée un nouveau modèle, jusqu'au résultat voulu.

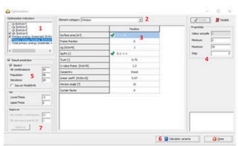
Dans nos tests, la convergence est assez rapide.

Application dans Lesosai – introduction des données

Ecran de démarrage:

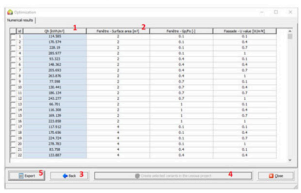


Choix des calculs et des variations sur les variables choisies:



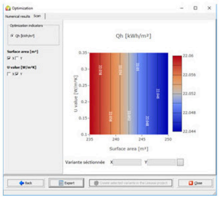
Application dans Lesosai – résultats

Résultats sous forme de tableau exportable:



Application dans Lesosai – résultats

Résultats sous forme de graphique pour analyses:



13 certifications & bilans écologiques et énergétiques de bâtiments E4tech software

Conclusion

Le modèle succédané développé dans le projet permet d'optimiser rapidement un bâtiment selon plusieurs points de vue (énergie, confort,...) et d'aider l'expert dans ses choix.

A la base, le projet est prévu pour le dimensionnement des fenêtres, mais il peut être appliqué à d'autres éléments de la construction (ventilation, valeurs U,...)

L'intégration dans Lesosai sera disponible dans la deuxième moitié de 2019. Des améliorations dans l'ergonomie sont en cours.

14 certifications & bilans écologiques et énergétiques de bâtiments E4tech software

Merci de votre attention

www.lesosai.com

www.e4tech-software.com

E4tech Software SA, Av. Juste-Olivier 2 – 1006 Lausanne, Suisse

certifications & bilans écologiques et énergétiques de bâtiments E4tech software

3

Optivitrage

Ein neues Werkzeug zur Verbesserung der Fenstereffizienz in Bezug auf Energie, Komfort und Größe.

Flavio Foradini
Windays 2019 - Biel

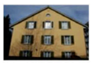
certifications & bilans écologiques et énergétiques de bâtiments E4tech software

Fragen

Wieviele Fenster soll ein Gebäude haben?

Welcher Fenstertyp?

Muss ich klimatisieren?


certifications & bilans écologiques et énergétiques de bâtiments E4tech software

Die Wirkungen von Fenstern

Die Wahl der Fenster / Vergrösserung der Fensterfläche hat Auswirkungen:

- Bezüglich Energiebedarf:
  - + Wärmegewinn
  - + Kühlbedarf
- Bezüglich technischen Installationen
- Bezüglich Komfort:
  - + Tageslicht
  - + Überhitzungsgefahr

certifications & bilans écologiques et énergétiques de bâtiments E4tech software

Klassische Arbeitsweise

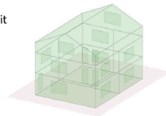
Der Architekt zeichnet den Entwurf und definiert mit dem Auftraggeber die Ziele (im Lastenheft).

Er schickt es an den Bauphysiker (Experten), der schnell versucht, Lösungen zu finden.

Das Projekt geht mit den Änderungsinformationen zum Architekten zurück.

Da die Arbeitszeit begrenzt ist, ist die Lösung, obwohl akzeptabel, oft nicht ideal.

Die Berechnung des Komforts in Grenzfällen erfolgt stündlich.



4 certifications & bilans écologiques et énergétiques de bâtiments E4tech software

Das Projekt **Optivitrage**

Ein Projekt des Bundesamtes für Energie.

Teilnehmer:

- E4tech Software AG: Lesosai Software und Kenntnisse der Normen
- Berner Fachhochschule, Fachbereich Architektur: Wissen über Fenster, Bauphysik und mathematische Methoden

Ziel des Architekten:

- ein Gebäude zu haben, das den schweizerischen Gesetzen entspricht und den Kunden befriedigt.

Ziel des Projekts

- Minimierung des Bedarfs an Primärenergie bei Beibehaltung des (thermischen) Komforts;
- bei gleichzeitiger Begrenzung des Zeitaufwands für die Berechnung und Lösungsfindung

5 certifications & bilans écologiques et énergétiques de bâtiments E4tech software

Arbeitsweise - 1

Der Architekt führt das Basisprojekt durch und definiert mit dem Auftraggeber die Ziele.

Mit dem Experten, definiert der Architekten den Handlungsspielraum (zum Beispiel) bezüglich:

- Konstruktionen: maximale Dicken der Elemente, minimale Stärken der Tragkonstruktion, Tragschichten, Lage der Materialien.
- Fenster: minimale und maximale Abmessungen, U-Werte,.....
- Belüftung: mechanisch / natürlich
- Beleuchtung: Grenzwerte und Sollwerte, Technik und Steuerungsart (z. B. manuell und automatisch)
- ...

6 certifications & bilans écologiques et énergétiques de bâtiments E4tech software

1

Arbeitsweise - 2

Der Experte führt eine große Anzahl von Berechnungen durch, analysiert sie und schlägt dem Kunden dann mehrere Lösungen vor. Diese können zum gleichen Ergebnis führen.

Beispiel Energiebedarf:
Es ist möglich, die Wände energetisch besser und die Verglasung energetisch weniger gut auszuführen, oder umgekehrt, und dabei das gleiche Ergebnis zu erzielen.

Methodik

Die stündliche Berechnung ist keine einfache Berechnung. Diese Methodik macht die Entwicklung eines alternativen Modells (sog. Ersatzmodells) notwendig, das sich dem unbekannten Modell annähert.

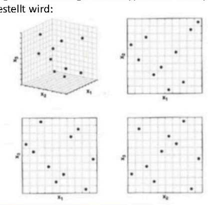
Schritte:

1. Vorbereitung einer Stichprobe von Parameterkombinationen
2. Entwicklung des alternativen Modells
3. Erforschung des des alternativen Modells, um ein oder mehrere Ergebnisse zu finden.
4. Mögliche Verbesserung des alternativen Modells

Methodik: Stichprobe

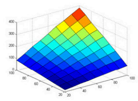
Der Zweck der Stichprobe ist es, den Raum zu füllen.

Die vorgeschlagene Lösung ist eine zufällige "Latin Hypercube"-Stichprobe, die durch die folgende Abbildung dargestellt wird:



Methodik: Ersatzmodell (alternatives Modell)

Basierend auf "realen" Berechnungen an der Stichprobe entwickelt das Verfahren ein Ersatzmodell, mit dem alle gewünschten Punkte schnell berechnet werden können:

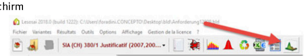


Tests werden verwendet, um die Qualität des Modells zu beurteilen. Wenn es nicht ausreicht, fügt die Methode einen weiteren Satz von Stichproben hinzu und erstellt ein neues Modell, bis das gewünschte Ergebnis erreicht ist.

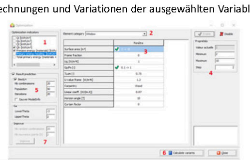
In unseren Tests ist die Konvergenz ziemlich schnell.

Anwendung in Lesosai – Dateneingabe

Startbildschirm

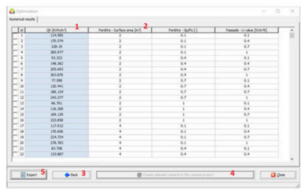


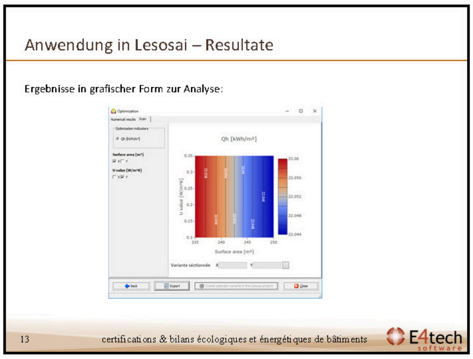
Auswahl der Berechnungen und Variationen der ausgewählten Variablen:

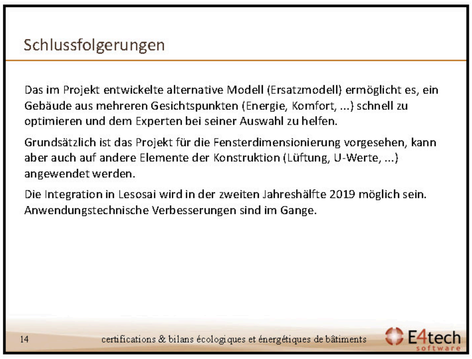


Anwendung in Lesosai – Resultate

Ergebnisse in Form einer exportierbaren Tabelle:




Anwendung in Lesosai – Resultate
Ergebnisse in grafischer Form zur Analyse:
Qh [kWh/m²]
Surface area [m²]
13
certifications & bilans écologiques et énergétiques de bâtiments
E4tech software


Schlussfolgerungen
Das im Projekt entwickelte alternative Modell (Ersatzmodell) ermöglicht es, ein Gebäude aus mehreren Gesichtspunkten (Energie, Komfort, ...) schnell zu optimieren und dem Experten bei seiner Auswahl zu helfen.
Grundsätzlich ist das Projekt für die Fensterdimensionierung vorgesehen, kann aber auch auf andere Elemente der Konstruktion (Lüftung, U-Werte, ...) angewendet werden.
Die Integration in Lesosai wird in der zweiten Jahreshälfte 2019 möglich sein. Anwendungstechnische Verbesserungen sind im Gange.
14
certifications & bilans écologiques et énergétiques de bâtiments
E4tech software


Vielen Dank für Ihre Aufmerksamkeit.
www.lesosai.com
www.e4tech-software.com
E4tech Software SA, Av. Juste-Olivier 2 – 1006 Lausanne, Suisse
certifications & bilans écologiques et énergétiques de bâtiments
E4tech software

3