

Part 1:	An Introduction
----------------	------------------------

1.1 Application Information

WAEBRU analyses temperature conditions in building components in which heat flows not just in one direction, but in two- and three-dimensional patterns. This is of critical interest when evaluating the thermal performance - an important design criterium - of so-called *thermal bridges*, that is, regions of a building construction through which local peaks of heat loss can cause surface temperatures to drop below the dewpoint. Thermal bridges are characterised by multi-dimensional heat flow, and therefore by the fact that they cannot be adequately approximated by the one-dimensional models of calculation typically used in norms and standards for the thermal performance of buildings (U-values).

The method of analysis most commonly practiced today when evaluating the thermal performance of building spaces, components, and assemblies is based on a simple, one-dimensional, constant flow model of heat conduction (i.e. the assumption of parallel heat flow for the calculation of U-values and areas). Such an assumption often leads not only to disappointing results in the thermal performance of realised construction projects, but also to costly consequences due to unforeseeably high energy consumption for heating, as well as damage caused by surface condensation of moisture.

These potentially negative consequences of over-simplification, inherent to the assumption of one-dimensionality, are becoming increasingly critical in today's trend towards highly insulated building structures. If the effects of thermal bridges are neglected, drastic errors in estimating heating requirements are bound to result, particularly when assessing energy efficient buildings. Multi-dimensional (i.e. two- and three-dimensional) evaluations of thermally critical regions within a building assembly during early design phases can provide valuable preliminary information to support the decision-making process, thus leading to considerably more reliable design results.

Surface moisture due to condensation (typically occurring in such regions as floor-wall connections, window installations, etc.) as well as mould growth in humid environments can also be effectively prevented by means of multi-dimensional evaluation during planning and detail design.

Drafts of the European Standards (prEN) pertaining to such aspects of thermal performance of building constructions are already available¹. The ramifications of the adoption of these European Standards is apparent: they describe the substance of new tasks for designers, manufacturers, and builders, as well as their significance for quality control in the field of energy conservation. For a more detailed description of the concepts relevant to evaluating thermal bridges, see Chapter 1.2, **Primary Concepts**.

The purpose of the standard prEN "Thermal Bridges - Calculation of Surface Temperatures and Heat flows" is explicitly stated as:

- the calculation of minimum surface temperature in order to assess the risk of surface condensation and
- the calculation of heat flows (for the constant flow case) in order to predict overall heat loss from a building.

Furthermore, "high precision" methods of calculation are demanded, whereby precision criteria which must be satisfied by the method chosen are defined in the standard.

¹ prEN 32573, "Thermal Bridges in Building Construction - Heat Flows and Surface Temperatures - General Calculation Methods", 1992.

prEN 26946-1, "Building Components and Building Elements - Thermal Resistance and Thermal Transmittance - Calculation Method", 1992.

According to prEN "Building Components or Building Elements - Calculation of Thermal Transmittance", calculation of heat transfer coefficients of parallel plane surface building components shall be performed based, of course, on one-dimensional models. For such components, composed of surface parallel layers of thermally homogeneous or non-homogeneous materials, a well-known method of estimation (ISO-method) can be implemented in order to obtain a design thermal resistance - assuming the maximum relative error remains "negligible".

For all other heat flow patterns, i.e. for all cases of more than one dimension, in particular in the presence of thermal bridges, the standard prEN "Thermal Bridges" requires the implementation of numerical methods of evaluation.

The program package WAEBRU 5.0 meets the demands of the European Standards with respect to input data and - more importantly - output data requirements (evaluation results).

With respect to input data, WAEBRU

- facilitates the generation of geometrical models by supporting a graphic input display of building structures, as well as by providing an entirely independent, fully automatic method of fine subdivision (which can also be influenced by the user through manual manipulation of parameters).
- delivers complete input model documentation (geometry of material elements, thermal design values of materials, spaces, heat sources...) upon request.
- allows the precision of numerical solutions to be influenced and controlled by the user (definition of calculation parameters).

Evaluation results yielded by WAEBRU include

- generally applicable results in the form of g-values and conductance matrices. These conform to the temperature weighting factors and "thermal coupling coefficients" defined in the European Standards, including the required information on calculation precision.
- specific results, applicable to particular air temperature conditions in spaces thermally coupled by the building components analysed, in the form of surface temperature minima and maxima as well as respective dewpoints.
- graphic plots and prints of isotherms, surface temperatures or temperatures along an edge (2- or 3-dim. models), as well as heat flow diagrams (2-dim. models only).

Furthermore, fully automatic execution of calculation with WAEBRU is given even in the event of poorly conditioned calculation cases. The maximum quantity of balancable cells (with 16 MByte RAM) is approx. 240 000, thus making the thorough analysis of large, three-dimensional models feasible. Complex cases are given, for example, by

- components or spaces in contact with ground.
- entire spatial envelopes or groups of spaces.
- detailed modelling of complicated assemblies (window frame and installation details, steel structures, etc.).

The thermal performance of building assemblies with overall dimensions of up to app. 100 m can be simulated with WAEBRU. Limitations to the scope of evaluation depend on the PC hardware implemented.

1.1.1.**System Requirements**

The computer hardware necessary for running the program package WAEBRU successfully depends on the nature (and complexity) of the constructions to be analysed as well as on the quality of results precision sought by the user. Both factors affect the number of equations which must be solved in the course of calculation. This number is limited primarily by the amount of memory available. Therefore, if an emphasis on three-dimensional applications can be expected, a more powerful configuration than the minimum described here is advisable.

WAEBRU 5.0 has been programmed especially for use on 80386 or 80486 systems. The program package is able to operate with extended memory in addition to the conventional DOS memory through implementation of the OS/386 DOS-Extender by Ergo Computing Inc.

Depending on the extent of the calculation model, WAEBRU automatically selects the appropriate one of three possible operating modes:

1. For the calculation of small models, i.e. if less than approximately 7000 cells need to be balanced, the program runs conventionally in *real mode* (also referred to as *DOS-mode*).
2. In order to calculate a larger model, the extended memory must be utilised. In this case, the program operates in *protected mode*, whereby the entire memory available can be accessed linearly.
3. In the event that a calculation model is so extensive that the memory available in *protected mode* is not sufficient, WAEBRU accesses additional memory on the hard disk as necessary for operation.

The DOS-Extender, which allows program operation in protected mode (2 and 3), must be installed separately by the user before running WAEBRU for the first time. Once installation and appropriate configuration of the memory extender is complete, this module is automatically executed by WAEBRU 5.0 when needed for calculation. This has no effect on the general operating system, DOS.

The third calculation mode described above, which involves the use of temporary auxiliary files on the hard disk, leads to considerable increases in computation time as well as hard disk wear. It is therefore advisable to limit the operation in this mode to special cases.

The protected modes 2 and 3 are implemented by WAEBRU only in the calculation branch of the program package. The branches for input, evaluation, and graphic output run in conventional DOS-mode (1).

Hardware and Essential Software

The minimum configuration recommended for installing WAEBRU includes the following basic components:

- COMPAC 386, 386SX, IBM PS/2 Model 70 or a compatible 386 system (80386 CPU must be at least step B0). 80486 systems are also supported.

Note: Compatibility to the 386 standard is assumed. 386SX systems with incompatible ROM-BIOS, older 16 MHz 386 systems, or such with 386 add-in cards could result in compatibility problems.

- 80387 mathematic co-processor.

Note: This is already included in 486-DX systems, whereby the 80387 co-processor is integrated in the 80486DX chip (not valid for 486-SX systems).

- 640 kB conventional memory as well as at least 4 MB RAM for extended memory.

Note: Conventional memory is used in mode 1. As soon as this capacity is no longer sufficient, the DOS-Extender is loaded for operation in mode 2 with the extended memory. Calculations requiring even further capacity are run in mode 3 using memory from the hard disk. For the reasons already described above, this third mode should be avoided whenever possible.

Thus, if more extensive applications are to be expected regularly, it is advisable to upgrade RAM in order to improve operation within mode 2 (up to 16 MB).

- Hard disk with at least 40 MB memory.
- 3 1/2" floppy disk drive (1.44 MB).
- DOS operating system software from version 3.3 to 5.0.
- EGA graphic card (256kB memory) or VGA graphic adapter, and appropriate monitor.
- Printer (Epson FX-80 or better - see list below).
- Plotter (optional - see list below).

memory [kB]	max. no. of cells
640	ca. 7 000
5 000	ca. 70 000
8 000	ca. 110 000
16 000	ca. 240 000

As already mentioned, the maximum number of cells (model size) which can be calculated in mode 2 depends primarily on the memory capacity of the computer used. The table to the left shows approximate values, which were derived in a series of program test runs on various hardware configurations under the condition of minimal extraneous programs resident in memory.

The computation time to be expected in calculating a particular model depends on the speed (frequency) and grade of the processor(s) used. With a 486DX/33 MH processor, for example, the two-dimensional model described in the **Tutorial** of this manual was calculated in less than two minutes. The three-dimensional, extended model of the tutorial object required 57 minutes from calculation branch begin to return to the Main Menu.

A special UNIX version of the calculation branch of WAEBRU is also available for installation on a mainframe computer, if required.

80386 Chip Problems:

The "B-step" 80386 processor (used in earlier 16 MHz systems) as well as some 20 MHz processors have a fault which reveals itself only in connection with the 80387 co-processor. This error lies within the 386 CPU and not the mathematic co-processor. Should problems with the 80386/80387 chip set arise, please contact the hardware distributor.

Supported Output Devices:

Calcomp ColorMaster
 Canon BJ80
 Canon BJ-130E at 360 dpi
 Canon LBP-811 Laser Printer (vector mode)
 Corona Laser Printer
 C. Itoh Prowriter
 C. Itoh 24LQ
 Epson and compatible with 360 dpi
 Epson FX, RX
 Epson HI-80 plotter
 Epson MX
 Epson LQ1500
 Hewlett Packard HP-GL plotters
 Hewlett Packard InkJet (SW5 up) (6.5 x 8.5 in.)
 Hewlett Packard LaserJet+
 Hewlett Packard PaintJet in collar
 Houston DMP-xx plotters
 IBM Graphic or Professional
 Micro Peripherals 150, 180
 NEC Pinwriter P series
 NEC Pinwriter P series with colour ribbon
 NEC 8023, 8025
 Okidata ML 393 at 360 dpi
 Okidata 92, 93
 Okidata 192+ (eight bit graphics)
 PostScript (TM) printer 300 dpi
 Quadram QuadLaser (with vector software)
 Roland DXY-800 plotter
 Seikosha SK-300AI (no colour)
 Seikosha SK-300AI with colour ribbon
 Smith Corona D100
 Tektronix output code
 Tektronix 4695 ink jet printer
 Toshiba P321 and P351 (unidirectional printing)
 Toshiba P351C with colour ribbon
 Toshiba 1340 (not unidirectional)
 Watanabe Digi-Plot plotter

1.1.2.**Installing WAEBRU 5.0**

The program package WAEBRU 5.0 has been delivered on three 1.44 MB floppy disks (3 1/2"). One further disk contains demonstration files with which the examples illustrated in the **Tutorial** of this manual can be reviewed. A list of all program files can be found below.

Before beginning installation of WAEBRU, it is advisable to make a set of 3 1/2" backup disks containing copies of the original files (with the DOS command `diskcopy`). In the event that the `diskcopy` operation repeatedly leads to the error message

Error reading drive A

with a particular floppy disk, please request the delivery of a new original of this disk.

Once all the original program disks have been copied, these should be stored in a safe place and the backup disks used for all further procedures.

disk 1:

WAEBINP	EXE
WAEBOUT	EXE
WAEBPLOT	EXE
WAEBRU	HID
WAEBRU	INI

disk 2:

EQ_PREP	BAT	RATOSOR	BAT	SORF1	EXE
GE_DONE	EXE	RA_SOR	EXE	SORF2	BAT
GE_SOR	EXE	RESOLVE	BAT	SOR_OUT	EXE
OMEGAO	EXE	RHS_GEN	EXE	WAEBCLEA	BAT
OMEGAOF2	EXE	SOLVE	BAT	WAEBBAS	EXE
OMEGAOPT	BAT	SOR	EXE	WAEBRU	BAT
OUT_SOR	EXE	SORELAX	BAT	WAEBRU	ERR

disk 3:

OMEGAF2E	EXE	SORF1E	EXE
OMEGAOE	EXE	SORF2E	EXE
OS386	EXE	386SETUP	EXE
RA_SORE	EXE	386STAT	EXE
SORE	EXE	386TUNE	EXE

demo disk:

TUTOR2	201	TUTOR3	301
TUTOR2	202	TUTOR3	302
TUTOR2	20A	TUTOR3	30A
TUTOR2	2AC	TUTOR3	3AC
TUTOR2	2BT	TUTOR3	3BT
TUTOR2	2PA	TUTOR3	3PA
TUTOR2	2RA	TUTOR3	3RA
TUTOR2	2GR	TUTOR3	3GR

Installation on Hard Disk

WAEBRU requires 2.6 MB of hard disk memory for program files. Further memory is necessary for loading the demonstration (tutorial) files.

hard disk capacity

In the course of running WAEBRU, a number of potentially large, temporary files may be generated. Enough hard disk capacity must remain available for these files in order to ensure proper program operation.

A general rule of thumb is that the available hard disk memory should be at least double the RAM in use. This means, for example, that 16 MB should be reserved on the hard disk for an application running on a computer with 8 MB RAM. It is therefore advisable to check the situation on the hard disk before running WAEBRU to avoid unnecessary complications.

In the following directions, it is assumed that WAEBRU shall be installed on a single-station, personal computer.

copy

First a drive for the program must be determined and a directory created. Here, the directory name **waebru** on drive C has been assumed; please substitute the appropriate drive and name (path) in the command examples.

To switch to drive C, type

```
c:
```

and then establish a directory by entering

```
md c:\waebru
```

It is then necessary to change to this directory for subsequent copying:

```
cd \waebru
```

Insert disk 1 in drive A, and type

```
xcopy a:*. * /v
```

Once disk 1 has been thus copied into **waebru**, repeat the last step for disks 2 and 3.

The demonstration files on the remaining disk can be installed in the same manner. For the sake of clarity, it is suggested that these be copied into a subdirectory of **waebru**, e.g. **c:\waebru\demo**. The subdirectory is made by typing

```
md demo
```

and changed with

```
cd demo
```

Type the command

```
xcopy a:*. * /v
```

to install the demo disk on this path.

Once all the necessary WAEBRU files have been successfully installed on the hard disk, the system must be configured for optimal operation. These stages of installation are described in the next sections of this chapter.

Changes in the CONFIG.SYS File

In order to run WAEBRU 5.0, the following parameters must be defined in the system file **CONFIG.SYS** :

```
SHELL = COMMAND.COM /P /E: 1024
FILES = 20
BUFFERS = 30
```

In the event that these lines have not yet been included, they must be added here. If the command line **shell** already exists, the option parameter **/E:** must still be increased by 1024 (as illustrated above). For more information about commands of the system configuration file, please refer to a DOS manual.

Furthermore, **device** entries such as

```
DEVICE = HIMEM.SYS
DEVICE = SMARTDRV.SYS
DEVICE = FASTOPEN.SYS
```

may also have been included in the **CONFIG.SYS** file. These lines can, however, cause a slowing down of the entire system in connection with the DOS-Extender, and should therefore be eliminated (or at least deactivated with **rem**) before running WAEBRU.

Changes in the AUTEXEC.BAT File

As previously mentioned, the program package WAEBRU 5.0 utilises virtually all the memory available with DOS. Therefore, special attention should be paid to the availability of as much memory as possible by limiting the amount occupied by any resident programs. Certain resident programs may even impede execution of the DOS-Extender.

The programs automatically loaded at system start can be kept to a minimum through appropriate command lines in the **AUTOEXEC.BAT** file (see DOS manual). In order to indicate the existence of an extended memory for WAEBRU, it is absolutely necessary to include the following line in this file:

```
SET USE386 = YES
```

If WAEBRU should be accessible from other working directories, then the path on which the program package lies must also be specified with a command line here, e.g.

```
PATH = C:\waebru
```

Configuring the DOS-Extender

The DOS-Extender must be configured using the program **386TUNE** to ensure its proper installation on the given system. After changing to the directory of WAEBRU, the program file **386TUNE.EXE** is executed by typing the run command:

```
386TUNE
```

386TUNE

Attention:

The execution of 386TUNE can, under circumstances, lead to a system crash. This program should therefore never be run before the system has been set to allow re-boot (no BOOT-partition on the hard disk and no BOOT-disk).

386TUNE attempts to improve the performance of the DOS-Extender. It includes an internal data bank with different system types. When the system currently in use has been identified in this data bank, 386TUNE automatically configures the DOS-Extender to completion.

In the event that 386TUNE cannot identify the system with its data bank, it begins to run system tests. During the test phase, running test results must each be confirmed by pressing the space bar. If the program should stop reacting to these repeated confirmations, the system must be re-started. This procedure (system boot and re-execution of 386TUNE) could repeat itself up to four times, depending on the type of system being tested. 386TUNE records preliminary results in a special file so that testing can continue where it left off at the last break.

Once 386TUNE has been executed successfully (without aborting), configuration of the DOS-Extender is complete. This is indicated with a message in the following form:

```
      Please upload file bda147d3.db to our BBS.  
(508) 535-7228 1200/2400 baud, 8 data bits, 1 stop bit
```

The file mentioned here, with the extension **db**, has been generated by the DOS-Extender. It is required for WAEBRU calculations.

386SETUP

Further parameters for the DOS-Extender can be defined by means of the program **386SETUP.EXE**. However, this program should only be implemented by users who are sufficiently experienced in system configuration. The following suggested settings for the DOS-Extender can be specified with the aid of 386SETUP:

```
LOWALLOCATE = 1000  
LOWRESERVE  = 100
```

Configuring the WAEBRU Program Package

Configuration data for WAEBRU is specified in the initialisation file **WAEBRU.INI**. Parameters pertaining to the four program branches of WAEBRU (input, calculation, evaluation, and output), as well as to peripheral devices can be edited here using any appropriate text editor.

An example version of **WAEBRU.INI** is delivered with the program package; this file is illustrated at the beginning of Appendix A - **Configuration Data**.

The various initialisation commands are described in detail in the appendix sections

- **Peripheral Parameters,**
- **Program Parameters, and**

- **Calculation Parameters for the Relaxation Method** (see also **Numerical Solution** in the next chapter - **Primary Concepts**).

1.2. Primary Concepts

The various terms and theoretical concepts, which form the basis for understanding the WAEBRU application, are presented and defined briefly in the following sections of this chapter. For a more thorough explanation of the theoretical background, please refer to the book "Wärmebrücken"² (available upon request).

1.2.1.**Environmental Factors**

When analysing the thermal performance of building components, two different types of information are sought: thermal transmission and surface temperatures.

The first type entails evaluating the heat transfer characteristics of a building assembly, i.e. the form and quantity of heat losses and gains, in order to ultimately obtain

- a reliable estimation of heating and cooling loads (costs),
- acceptable heat flow rates,

as well as

- energy conserving design alternatives.

Based on the heat transfer characteristics of a construction, the expected temperatures along interior surfaces must be evaluated in order to predict (and avoid) areas of potential moisture condensation. Beyond preventing damage to building materials caused by mould growth, adequate surface temperatures are also a relevant factor in the thermal comfort of an interior environment.

Thermal Transmission

Energy in the form of heat in building constructions is generally transmitted by a combination of conduction, convection, and radiation. When considering heat transfer within and through solid (homogeneous or non-homogeneous) building materials, *heat conduction* is the primary transmission factor; the effects of convection and radiation are typically negligible.

heat conduction

Conduction through homogeneous planar building components (i.e. composed of one or more layers of material with parallel surface planes) occurs in a single direction: normal to the component surface. This is referred to as *one-dimensional heat flow*, and is characterised by a constant surface temperature over the entire surface plane.

Such idealised conditions can, of course, only be assumed in limited regions of an actual building structure. Geometries of non-planar components (construction joints, floor-wall connections, balconies, etc.) give rise to heat flow patterns of more than one direction, that is, to *two- or three-dimensional heat flow*.

² W. Heindl et al., **Wärmebrücken**: Grundlagen, einfache Formeln, Wärmeverluste, Kondensation, 100 durchgerechnete Beispiele. Vienna: Springer Verlag, 1987.

thermal bridge

Regions or elements of a building construction characterised by multi-dimensional heat flow patterns are called *thermal bridges*. In contrast to regions of one-dimensional heat flow, thermal bridges are typically associated with local peaks of heat loss, which correspond to characteristic drops in the interior surface temperature.

Conductance

An exact description of the thermal behaviour of a building assembly would by nature be non-linear, and thus very complicated to evaluate. Fortunately, for most construction situations of interest, it is possible to drastically reduce the complexity of the physical model without sacrificing any appreciable accuracy.

A linear description of thermal transmission involving an entire building can be introduced by

- assuming that the temperature of each environment (space) associated with the building is unique, i.e. independent of position, and
- forfeiting an exact treatment of radiation exchange within the spaces in favour of an approximation of this factor through adjusted surface transfer coefficients.

thermal conductance

In light of this simplified physical model, it can be stated that the amount of heat which flows from one space to another, Q , is proportional to the temperature difference between the two environments under consideration (i and j). The factor of proportionality is a conductance³, L_{ij} , and linearly defined by

$$Q = L_{ij} \cdot (T_i - T_j) .$$

conductance matrix

The analogy of an electrical circuit can be extended to model a building structure more generally as a set of environments thermally connected through a resistance, i.e. the heat-conducting material elements. A complete description of the thermal relationships in a particular structure is given by a *conductance matrix* of values for all thermal connections, $i \cdot j$, between the spaces of the model.

The conductance matrix defines the heat transfer characteristics of a given model based solely on the geometry and materials (thermal properties) of the resistance, that is, independently of temperature conditions in adjacent spaces. This matrix is symmetrical.

thermal transmittance

The area-related term "thermal transmittance" (U -value in $\text{Wm}^{-2}\text{K}^{-1}$) commonly used in standards to date essentially describes the same conductance as the reciprocal of the sum of resistances of a planar component, that is, resistances in "series":

$$1/U = 1/\alpha_i + R + 1/\alpha_e$$

³ In the European Standards, this conductance is called the "thermal coupling coefficient".

whereby α_i and α_e are the surface transfer coefficients of the interior and exterior environments, and R represents the sum of material resistances of j constituent component layers. The resistance of an individual homogeneous (isotropic) layer is directly proportional to layer thickness, d , and indirectly proportional to the material conductivity, λ :

$$R_j = d/\lambda$$

However, this simple formula applies only to the planar regions of a building assembly in which strictly one-dimensional heat flow can be reasonably assumed.

length-related conductance

A further type of special situation is given where two-dimensional heat flow patterns are to be expected. Such a region is a stretch of the building assembly which can be evaluated with respect to a two-dimensional section - under the assumption that no heat flow occurs normal to the section plane. In this case, a length-related conductance, L^{2D} [$\text{Wm}^{-1}\text{K}^{-1}$], must be calculated for the applicable region.

The most general case, of course, is that of three-dimensional heat flow. For the regions of a construction in which no directional assumptions can be made about local heat flow patterns, only the evaluation of the (3D) conductance, L^{3D} [WK^{-1}], provides a reliable indicator of the thermal behaviour.

total conductance

Due to the linearity of conductances, an entire building can be modelled as a sum of parts, with each part evaluated according to the applicable geometric conditions. Of course, the summation of conductances is only applicable if the temperature difference is the same through all model parts (e.g. one interior and one exterior temperature valid for the whole building).

The model is sub-divided by introducing theoretical cut-off planes, which must be located such that any potential heat flow through these planes can be considered negligible.

The *total thermal conductance* of a building thus modelled can be written as

$$L = \sum_i L_i^{3D} + \sum_j L_j^{2D} \cdot l_j + \sum_k U_k \cdot A_k$$

whereby l_j is the length over which the two-dimensional conductance, L_j^{2D} , is valid for part j , and A_k is the area of validity for U_k .

This reliable and flexible approach to analysing the thermal performance of buildings is referred to in the European Standards as the **direct method**. It requires the implementation of a suitable computer program for attaining two- and three-dimensional conductance results with the necessary precision.

The program package WAEBRU makes use of the linear nature of the heat conduction model by first determining a generally applicable calculation model: a characteristic set of temperature-independent *base solutions* (see the next section on **Method of Analysis**).

Surface Temperatures

Once the base solutions describing a particular model have been established, temperature (boundary) conditions can be applied in the adjacent environments in order to predict a specific temperature distribution in the building component.

dewpoint

Here the temperature along the component surface - where condensation is most likely to occur - is of particular interest. Humidity exceeding the saturation level of an environment, i.e. the *dewpoint* [%], leads to condensation. This value is directly linked to the temperature at any given location under.

temperature extremes

Therefore surface points at which a temperature minimum can be expected to occur must be localised in the model and explicitly evaluated with respect to specified conditions:

- boundary conditions - air temperatures [°C]
- position of coldest surface point (x, y, and z coordinates)
- temperature of the surface at this point [°C]
- associated dewpoint [%]

Note: Temperature maxima, i.e. warmest surface points, are primarily of interest when assessing the factor of thermal comfort in constructions which include *heat sources*. For example, situations in which electrical heating could be implemented (as floor heating, or as a measure against condensation) may be evaluated comparatively to determine if the power output needed could lead to uncomfortably high surface temperatures.

Building elements which absorb and re-emanate solar gain can also be effectively approximated as heat sources and thus evaluated in order to assess constructions critically affected by solar radiation.

Weighting Factors

The air temperatures set in the spaces adjacent to a building component all contribute to the temperature distribution resulting within the component. Therefore, the specific temperature, T , at any given point in a model can also be described as the result of all the space temperatures, T_0 through T_n , weighted for the specific position and summed:

$$T = g_0 \cdot T_0 + g_1 \cdot T_1 + \dots + g_n \cdot T_n$$

g-values

The set of weighting factors, g_0 through g_n , must be determined for each point of the model to be considered more closely. These so-called *g-values* are normalised such that their sum is equal to one.

If all the interior spaces are set at the same temperature (T_i for T_1 through T_n), and T_0 is defined as the exterior temperature (T_e), then the equation above can be simplified to

$$T = g_0 \cdot T_e + (1 - g_0) \cdot T_i ,$$

and re-written as

$$T = T_i - g_0(T_i - T_e).$$

In this case, g_0 represents a generalised version of the f -value familiar from the evaluation of surface temperatures for one-dimensional heat flow (based on U -values). Contrary to a conventional f -value, which applies to the entire interior surface plane, the g -value above is only valid for a specific point of a thermal bridge.

Once g -values have been characterised for the coldest points of all interior surfaces, however, the temperatures resulting at these points can be evaluated as simply as with the one-dimensional method.

Graphic Representation of Heat Flow

Since the heat flow pattern of a thermal bridge is characteristically more complex than the single-direction flow through a planar component, graphic illustrations of bridge flow patterns can provide critical information at a glance.

heat flow diagram

One method of visualising heat flow is to delineate the direction of flow (vector) through the building component with lines drawn at prescribed intervals. The area bounded by two lines in the diagram represents a heat quantity defined by the interval (e.g. given 10 intervals, the heat flow between two lines corresponds to 10% of the total heat flow in one direction through the surface of a given space). The denser the lines in a region of such a diagram, the more heat flows through this region. Thus local peaks of heat loss (and predictably cold surface points) are easily discernible.

Due to the two-dimensionality of graphic illustrations, however, three-dimensional heat flow patterns cannot be meaningfully rendered with such a diagram.

isotherms

The second common method of heat flow representation is to render the temperature distribution in a component with isotherms, i.e. by delineating lines of the same temperature at defined intervals. Isotherms lie normal to the direction of heat flow, thus providing an "inverted" rendering of the heat flow pattern (denser isotherms correspond to regions of increased heat flow).

Since isotherms represent a two-dimensional section through a temperature distribution rather than vectors directly, the isotherm method is also suited for (partial) rendering of three-dimensional heat flow situations.

1.2.2.

Method of Analysis

Heat flow in a building construction, as well as the subsequent temperature distribution, can be described mathematically by differential equations. Most importantly, these equations are linear and homogenous by nature.

base solutions

This means that one set of solutions, calculated for a specific set of conditions, can be "re-used" as the basis for solutions under differing conditions by *superposition*, i.e. by linear combination of selected *base solutions*. More specifically, base solutions are calculated under the assumption of a "basic" set of boundary conditions: with an air temperature of 1 in the selected space, and 0 in all others.

Thus the temperature distribution, a function of the three spatial coordinates, can ultimately be written in the simple form of a sum of temperature values. The individual temperatures are the result of the actual boundary conditions (air temperatures in the given spaces from 0 to m) weighted by dimensionless base solutions:

$$U(x,y,z) = \sum_{j=0}^m T_j \cdot g_j(x,y,z)$$

In other words, base solutions are effectively a generalised form of weighting factors (g -values) - a function of position for a given space, j : $g_j(x,y,z)$.

The calculation approach in the program package WAEBRU uses this circumstance to minimise over-all evaluation time. One set of base solutions need be calculated only once to characterise a given model, which can subsequently be considered under varying conditions without repeating the time-consuming computation necessary to solve the primary set of differential equations.

Computation time is further reduced by utilising the weighting function character of base solutions (normalised such that their sum must equal 1). Hence, if n cases have been selected, only $n-1$ solutions need to actually be calculated. The n -th base solution is then very simply derived as a difference of the sum to 1, that is, by a separate stage of superposition.

Calculation Model

Using the concept of a conductance, L_{ij} (between the spaces i and j), the heat which flows out of space i is a sum depending on the temperatures of space i and of all other spaces, j :

$$Q_i = \sum_j L_{ij} \cdot (T_i - T_j)$$

Since L_{ij} can be proven to be symmetrical ($= L_{ji}$), this equation corresponds formally to Kirchhoff's law of electrical networks. The analogy that this allows is quite significant: the spaces of a building construction (model) can be thought of as nodes in a heat conducting network.

equation model

Thus the heat conduction model, otherwise only describable with partial differential equations, can be mathematically defined as a relatively simple, linear system of equations with a given conductance matrix. Furthermore, only the thermal coupling between spaces which interchange heat directly through the building component ("neighbouring spaces") is of practical effect; coupling between indirectly connected spaces is generally negligible.

conditions

As the equation above shows, in addition to the appropriate value in the conductance matrix, either air temperature or heating load must be defined for each space of the model (evaluation conditions) in order to attain a complete description of a particular heat flow situation.

Numerical Solution

The calculation method of finite differences used by WAEBRU is based on the sub-division of the heat conducting continuum into an orthogonal cell structure. Each cell of this calculation geometry is treated as a *node* of the thermal network, and is typically connected with six other such nodes. The conductance of each connection depends on cell size and thermal conductivity of the material "filling" the individual cells. Therefore, the cell structure must be defined such that each cell can be assigned a single, homogeneous material.

calculation geometry

The calculation geometry is obtained by means of *gridding* the construction geometry of the model to be evaluated. A concrete grid is generated by dividing the model with a series of *grid planes* along material boundaries and at additional intervals (their number and distance depends on the precision of analysis required) parallel to the coordinate axes.

Once the network data has been sufficiently defined for calculation, the program sets up an appropriate linear system of equations automatically. This equation system then needs to be solved numerically with an iterative method.

relaxation factor

The convergence process of the relaxation method used by WAEBRU is entirely dependant on the value of the relaxation factor ω . Initially this factor is defined only within a given range:

$$1 \leq \omega < 2$$

The "optimal" ω -value is defined as the constant value of ω which leads to the most expedient convergence of the calculation method. This value differs from case to case and is initially unknown.

Once the system of equations describing the model has been determined, the first stage of calculation is therefore an analysis of this system, followed by an iterative calculation to approximate a preliminary "optimal" ω -value, ω_0 . This stage can be influenced by two parameters defined in the **WAEBRU.INI** file:

- The termination condition for determining ω_0 is satisfied when the absolute difference between last value of ω_0 just calculated and the "old" value from the previous iterative step falls below a limit for which ω_0 is considered solved. This limit is defined in the line **OMEGAO_DELTA=** of the **WAEBRU.INI** file.
- A further termination condition is given by the maximum number of iterative steps which the program allows before stopping calculation and accepting the last value of ω_0 as optimal for the next stage of calculation. This step number limit is defined in the line **OMEGAO_STOP=**; the standard setting is 30.

A more precise approximation of ω_0 (e.g. by setting a more stringent termination condition or increasing calculation time) is usually unnecessary, since ω_0 just serves as an initial parameter for the relaxation factor ω , which is then modified with each iteration in the course of the second stage of calculation.

The range of ω variation,

$$\omega_{\min} \leq \omega \leq \omega_{\max},$$

depends on ω_0 for the lower limit according to the following equation:

$$\omega_{\min} = \omega_0 - k \cdot (\omega_0 - 1) \cdot (\omega_0 - 2),$$

whereby k is a pre-defined parameter between -1 and 0.

Since k is negative ($\omega_{\min} < \omega_0$), specifying a greater absolute value of k results in a larger difference between ω_{\min} and ω_0 , and thus a larger range. This parameter is set in the line **OMEGA_MIN=** of the **WAEBRU.INI** file; the standard value is **-0.25**.

The upper limit of the range of ω variation, ω_{\max} , is a set value between ω_0 and 2. This is defined in the line **OMEGA_MAX=**, for which **1.998** is the standard setting.

In the second stage of calculation, the first step is performed with $\omega = \omega_{\min}$. The relaxation factor is then increased incrementally with each iteration until either ω_{\max} is reached or the process begins to diverge, whereupon ω is set back to ω_{\min} and calculation continues.

The increments of ω variation also vary: low values of ω in the beginning are run through rapidly, but the increment approaches 0 asymptotically as ω approaches the value of 2. The increments are calculated such that the approximated optimal value of ω_0 is reached within a given number of iterations starting from ω_{\min} . The standard number of assumed steps is 3; in the **WAEBRU.INI** file, this number appears transformed as a parameter of negative value in the line reading **OMEGA_WEIGHT=-3**.

As already mentioned, ω is set back to ω_{\min} as soon as the iterative results begin to recognisably diverge. The criterium for discerning whether the solution process is converging or diverging is the absolute value, Δ_{\max} , i.e. the deviation of the results from one iteration to the next. Hereby a *mean value* of Δ_{\max} is calculated from the last n steps and compared continually with the current Δ_{\max} . The number of steps included in the mean value for comparison, n , is also defined as a parameter in the **WAEBRU.INI** file: **OMEGA_TESTNUM=**, for which the standard setting is **10**.

The calculation process is considered divergent when Δ_{\max} equals or exceeds the comparative mean value. However, it would be impractical to set the relaxation factor back automatically every time this condition is satisfied before a certain minimum number of calculation steps has been performed. This quantity is a further criterium which must be met before an ω set-back occurs. The standard minimum number of iterations here is 23, defined in the line `OMEGA_VETO=`.

An equation is ultimately considered solved when the deviation, Δ_{\max} , remains smaller than a defined limit for a continuous series of a prescribed number of iterations. This quantity is defined in the line `TERM_NUM=` of the `WAEBRU.INI` file; the standard value is 10.

Finally, in order to "smooth" the results of calculation, a post-run of iterations is performed with a constant relaxation factor, $\omega = 1$ (defined in the line `OMEGA_POSTRUN=1.0`). An increase in this parameter should be avoided so as not to jeopardise the smoothing effect of post-calculation. The number of iterations of this stage is prescribed in the line `POSTRUN=`; standard is 15.

results precision

In summary, three primary factors control the precision of calculation results:

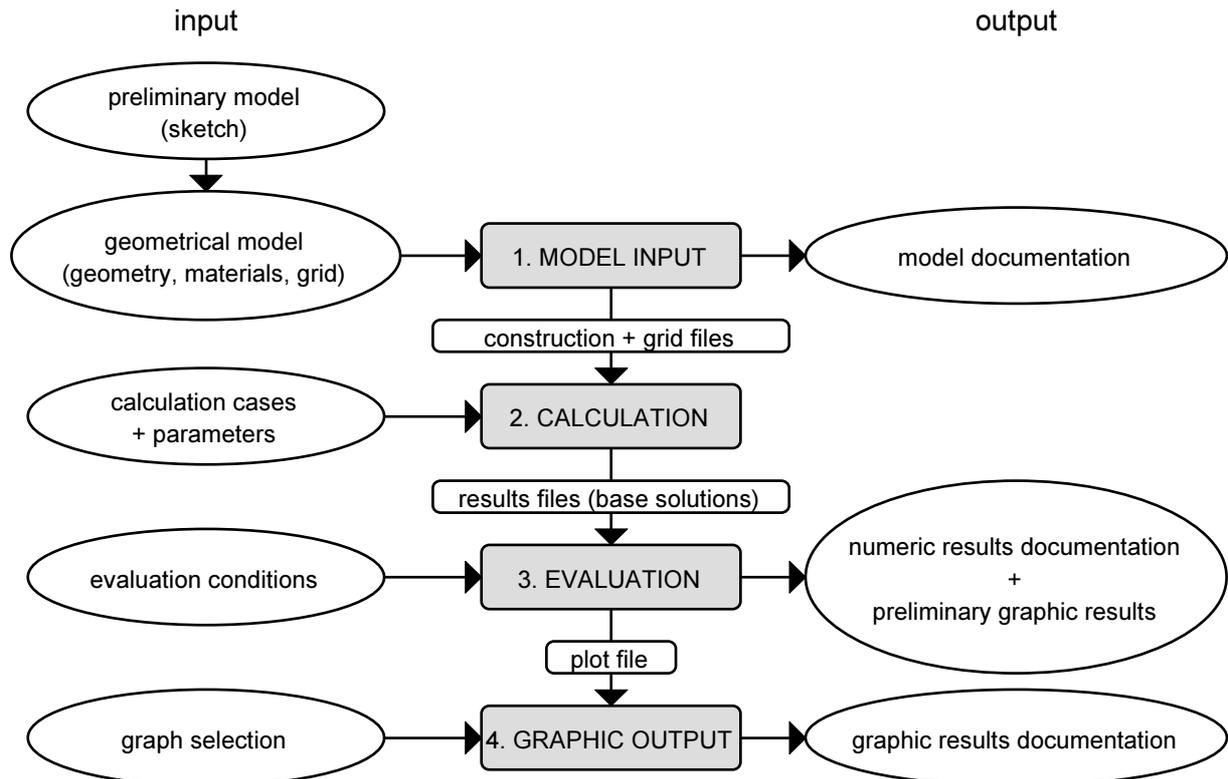
- fineness of the grid (calculation geometry).
- stringency of the termination condition (calculation time).
- precision of computation ("double precision").

The first two of these factors can be manipulated by the user, though the standard parameters used by WAEBRU are adequate for the evaluation of most models to be considered.

1.3. Structure of WAEBRU

WAEBRU is a package of numerous programs which are grouped in four major branches and coordinated centrally. The diagram below shows the over-all scheme of program operation. Stages of the primary sequence are connected with solid-line arrows. Potential secondary sequences - for editing and re-evaluating a given model - are not shown explicitly in the diagram.

The individual program branches are briefly described in the following sections of this chapter. A detailed explanation of the operation of these four branches is given in the next part of the manual, **Using WAEBRU**.



1.3.1.

Model Input

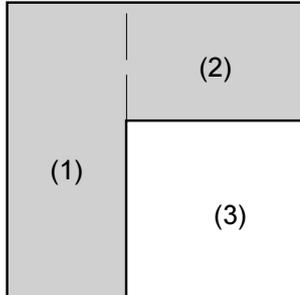
The input branch of WAEBRU supports the generation of a calculation geometry suitable for describing the object under consideration. The first stage of input entails generating a model of the object with respect to its geometry and the thermal properties of its materials.

The model is built up of *elements*, which are entered sequentially as rectangles defined by

- size and position in space - x and y coordinates (+ thickness in the z direction for three-dimensional models)

and

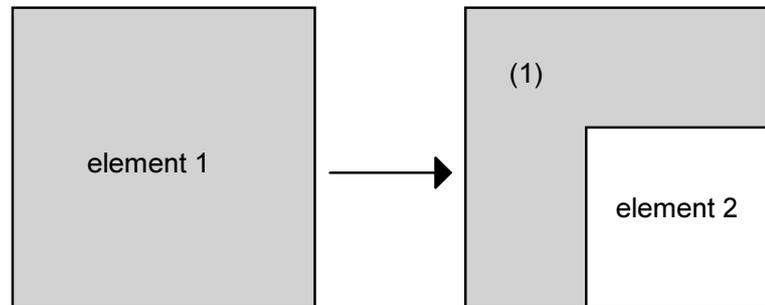
- material - type, identification, and relevant properties.

element overlapping

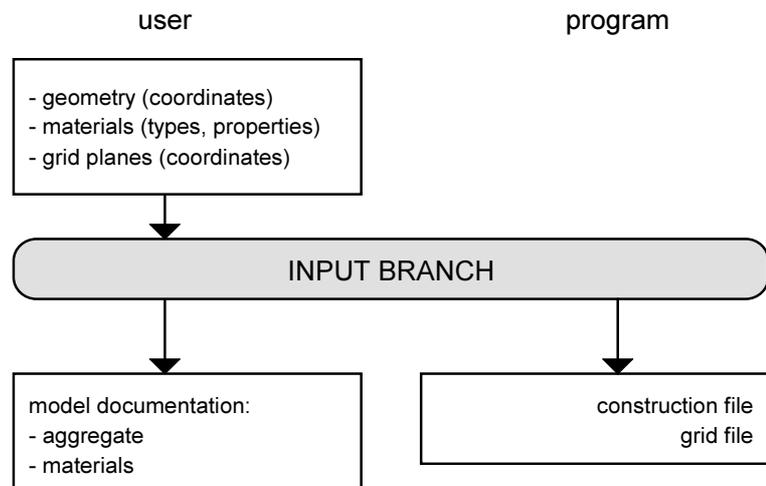
Elements of a shape more complex than a simple rectangle can be built up of sub-elements of the same material according to an additive principle. However, in many situations, a simpler subtractive principle inherent to WAEBRU can be implemented to achieve the same geometry more easily: *element overlapping*.

The principle of element overlapping means that any element entered "displaces" preceding elements (or any parts thereof) which share the same region in space. Which element overrides the other depends solely on the sequence of element input, as reflected in the element numbers.

Thus, for example, a shape such as the corner situation illustrated to the left can be drawn either as a composite of three elements or with a series of two over-lapping elements:



The second stage of input is the generation of the actual calculation geometry, that is, grid data based on the input model. Before continuing on to the calculation branch, two types of files must be generated which contain all the data necessary for a mathematical characterisation of the cell structure of the heat conducting model: construction and grid files.



1.3.2.

Calculation

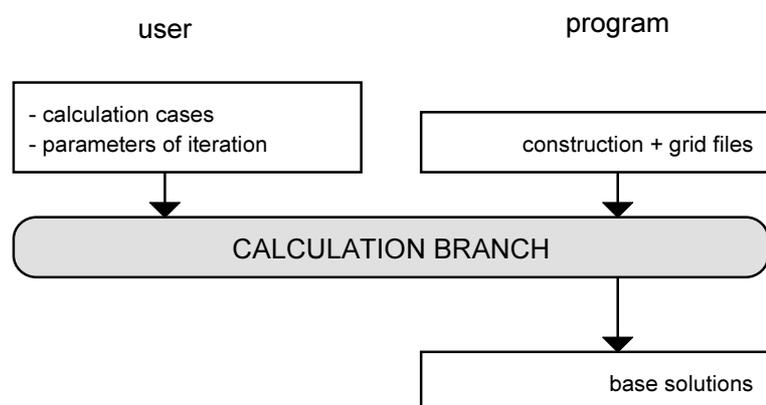
A characteristic of the program package WAEBRU is the division of the procedure for generating a temperature distribution into two stages: calculation of base solutions and evaluation under prescribed conditions. The first stage is connected with the numerical solution of a very large linear system of equations.

The number of equations in the system is equal to the number of cells in the orthogonal grid structure which it describes; thus a very finely gridded structure (calculation geometry) can result in lengthy computation. Herein lies the advantage of the "detour" over base solutions, since these need be calculated only once for a given model. The calculation time required for subsequent evaluation of specific temperature distributions under alternate boundary conditions is thus substantially reduced by employing the base solutions which have been determined in the calculation branch - instead of repeating the entire analysis for each set of conditions.

Before calculation can be started, cases for which base solutions are to be calculated must be explicitly selected (spaces, heat sources). This allows the user to further reduce calculation time in the event that spaces of the model can be combined and evaluated under the same temperature conditions.

The primary parameters of iteration can also be manipulated by the user prior to a calculation run. This is of interest when a model is particularly complex or when base solutions which have already been calculated prove to be inadequately precise during subsequent evaluation.

Once calculation is completed, the base solutions (as well as the status of results) are stored in results files for the model. These provide the basis of evaluation for the next branch of WAEBRU.



1.3.3.

Evaluation

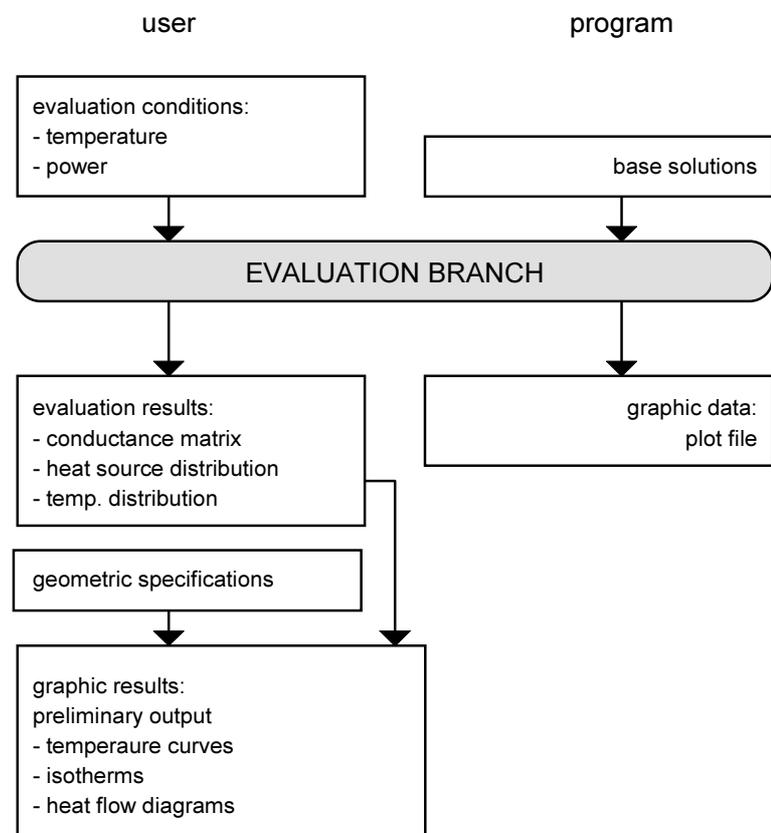
Final output data is obtained in the evaluation branch, during which the model under consideration (or rather, its characterisation through base solutions) is evaluated under specific conditions.

The boundary conditions which must be specified here include air temperatures for the spaces/cases calculated; the conditions at the outer boundaries of the model are already given by definition as so-called *cut-off planes*, i.e. as a non-conducting surface.

In the event that heat source elements have been included in the model, power values must be assigned as a condition to these cases as well.

Two types of output data are provided in the course of evaluation: numeric and graphic. Numeric data is printed here in the final form of evaluation results. Graphic representations of a given temperature distribution are generated in a form suitable for preliminary output in the evaluation branch. These can furthermore be stored selectively in a plot file for final output in the last branch of WAEBRU.

The various kinds of potential evaluation output are listed in groups in the diagram below.



1.3.4.**Graphic Output**

The output branch of WAEBRU is solely for viewing and/or plotting high resolution graphics of results obtained and written onto a model plot file in the evaluation branch.

Individual graphs (temperature curves, isotherms, heat flow diagrams) can be selected from an existing plot file for standardised output here.

