

In[589]:=

## Uživatelské nastavení parametrů ;

```
vyska=1; (* m; vyska vnitřní části zásobníku teple vody *)
objem=220; (* l; objem zásobníku *)
teplotaOhrivaneVody=55; (* °C; teplota, na kterou chci ohřívát vodu *)
minTeplotaOhrivaneVody=50; (* °C; minimální přípustná teplota teple vody *)
teplotaStudeneVody=15; (* °C; teplota neohřáté vody, která vteče do zásobníku *)
objemTeple=220; (* l; kolik je v zásobníku na začátku teple vody *)
vykonBojleru=2400; (* W *)
periodaSpinani=30000; (* s *)
c=4181; (*  $\frac{\text{J}}{\text{kg}\cdot\text{K}}$  *)
rho=990; (*  $\frac{\text{kg}}{\text{m}^3}$  *)
alpha=1.5*10^-7; (*  $\frac{\text{W}}{\text{m}\cdot\text{K}}$  *)
OffsetDown=0.9; (* udává, kolik může být maximálně studené vody v zásobníku *)
```

In[602]:=

## Parametry a deklaráce nutné pro chod programu (nezadává uživatel);

```
$HistoryLength = 2;
SetDirectory[NotebookDirectory[]];
odbery = Import["odbery.xlsx"][[1]];
teploty = Import["teploty3.xlsx"][[1]];
poloha =  $\frac{\text{objem} - \text{objemTeple}}{\text{objem}}$ ;
(* vzdálenost rozhraní tepla a studené vody od dna zásobníku v metrech *)
ohratCele = 0;
hladina =  $\frac{\text{objem} - \text{objemTeple}}{\text{objem}}$ ; (* počítáno od dna zásobníku <0;
1>, pomocná proměnná *)
funkce1 = 20 * Tanh[5000 * (x - hladina)] + 35;
If[objemTeple == objem, body = {#, 55} & /@ Range[0, 1,  $\frac{1}{50}$ ],
  body = {#, funkce1} & /@ Range[0, 1,  $\frac{1}{50}$ ]];
funkce = Interpolation[body];
den = 1;
fceTeplot = {};
dodanaEnergieDen = {};
dodatTepla =
  10^-3 * (objem - objemTeple) * c * rho * (teplotaOhrivaneVody - teplotaStudeneVody);
Q = 0; (* J *)
Qtotal = 0;
prumer = 2 * Sqrt[ $\frac{\text{objem} * 10^{-3}}{\text{Pi} * \text{vyska}}$ ]; (* metr *)
```

In[617]:= **Funkce pro zobrazení grafů;**

```
(* zobrazí rozdělení teploty vody v podélném směru zásobníku
vstup:
vystup: *)
Zobraz[] :=
  Show[Plot[funkce[x], {x, -0.1, vyska}, PlotRange → All, GridLines → Automatic,
    AxesLabel → {"výška zásobníku [m]", "teplota vody [°C]"}],
    Plot[minTeplotaOhrivaneVody, {x, 0, vyska}, PlotStyle → Red]];
```

In[619]:= (\* zobrazí dostupné množství teple vody v case

```
vstup:
vystup: *)
ZobrazfceTeplaVoda[] := Plot[fceTeplaVoda[x * 3600], {x, 0, 24},
  GridLines → Automatic, AxesLabel → {"čas [hod]", "objem teplé vody [l]"},
  PlotLabel → "Objem teplé vody v čase", PlotRange → {0, 225}];
```

In[620]:= (\* zobrazí v case kolik tepla je nutné

```
dodat pro ohrátí veskere vody na požadovanou teplotu
vstup:
vystup: *)
ZobrazfceDodatTepla[] := Plot[fceDodatTepla[x * 3600] * 10-6, {x, 0, 24},
  GridLines → Automatic, AxesLabel → {"cas [hod]", "chybejici teplo [MJ]"},
  PlotLabel → "Teplo pro ohrev vody v case", PlotRange → All];
```

In[621]:= (\* zobrazí odběry (objemový průtok) v case

```
vstup:
vystup: *)
ZobrazfceOdbery[] := Plot[fceOdbery[x * 3600], {x, 0, 24},
  GridLines → Automatic, AxesLabel → {"čas [hod]", "objemový průtok [l/s]"},
  PlotLabel → "Odběr teplé vody v čase", PlotRange → All];
```

In[622]:= **Funkce řešící rovnici vedení tepla;**

```
(* řeší rovnici vedení tepla
vstup: funkce určující rozdělení teploty vody napříč zásobníkem, čas po který probíhá p
vystup: řešení rovnice tepla *)
```

```
VyresVedeniTepla[vstupnifunkce_, tend_] := Module[{heq, poc, okr1, okr2, sol},
  heq = D[T[x, t], t] ==  $\alpha$  * D[T[x, t], {x, 2}];
  poc = T[x, 0] == vstupnifunkce[x];
  okr1 = D[T[x, t], x] == 0 /. x → 0;
  okr2 = D[T[x, t], x] == 0 /. x → vyska;
  sol = Quiet[NDSolve[{heq, poc, okr1, okr2}, T[x, t], {x, 0, vyska}, {t, 0, tend}, MaxStepSize →  $\frac{\text{vyska}}{100}$ 
  {sol}];
```

```

In[624]:= (* posunuje rozhranim teple a studene
vody na zaklade odberu/ohrevu vody v zasobniku
vstup: reseni rovnice tepla, cas trvani odberu/ohrevu vody,
vzdalenost o kterou se posune rozhrani mezi teplou a studenou vodou v metrech
vystup: *)
OdberOhrev[sol_, tend_, posun_] := Module[{body, body0, body1, body2},
  body0 = {# + posun, teplotaStudeneVody} & /@ Range[- vyska, -0.01,  $\frac{1}{100}$ ];
  body1 = {# + posun, T[x, t] /. sol /. t -> tend /. x -> #} & /@ Range[0, vyska,  $\frac{1}{100}$ ];
  body2 = {# + posun, teplotaOhrivaneVody} & /@ Range[vyska + 0.01, 2 * vyska,  $\frac{1}{100}$ ];
  body = Join[body0, body1, body2];
  funkce = Interpolation[body];
];

In[625]:= (* zjistí kde se nachází rozhraní mezi teplou a studenou vodou,
tzn voda o teplotě minTeplotaOhrivaneVody
vstup:
vystup:
vzdalenost rozhraní teple a studene vody od dna zasobniku v metrech *)
NajdiMnozstviStudeneVody[] := Module[{interval},
  interval = 0.1;
  poloha = 0;
  While[interval >= 10^-5,
    While[funkce[poloha + interval] < minTeplotaOhrivaneVody,
      poloha += interval;
    ];
    interval /= 10;
  ];
  poloha = If[poloha < 0, 0, poloha];
  poloha = If[poloha > 1, 1, poloha];
  {poloha}];

```

```
In[626]:= (* pocita mnozstvi teple vody v litrech a teplo,
ktere je nutne dodat studene vode aby se ohrala,
pridava hodnoty objemu teple vody do Listu objemTepleVodyVCase
vstup: zacatek zkoumaneho obdobi,
konec zkoumaneho obdobi, odber/ohrev teple vody v litrech
vystup: funkce urcujici rozdeleni teploty vody napric zasobnikem,
teplo ktore je nutne dodat studene vode,
aby se ohrala, objem dostupne teple vody v litrech *)
PrestupTepla[tstart_, tend_, odber_] := Module[{posun, sol},

If[tend > tstart, {posun =  $\frac{4 * odber}{10^3 * Pi * prumer^2}$ ;

sol = VyresVedeniTepla[funkce, tend - tstart][[1, 1]];
OdberOhrev[sol, tend - tstart, posun];
NajdiMnozstviStudeneVody[];
dodatTepla = c * rho *  $\frac{Pi * prumer^2}{4} * (teplotaOhrivaneVody * poloha -$ 
NIntegrate[funkce[x], {x, 0, poloha}, AccuracyGoal -> 3]);
objemTeple =  $\frac{Pi * prumer^2}{4} * (1 - poloha) * 10^3$ ;
objemTepleVodyVCase = Append[objemTepleVodyVCase, {tend, objemTeple}];
dodatTeplaVCase = Append[dodatTeplaVCase, {tend, dodatTepla}]}];

{funkce, dodatTepla, objemTeple}];
```

In[627]:= **Jednotlivé varianty ohřevu a odběru vody ;**  
**Varianta 1 – přímý ohřev kdykoliv je to možné;**

```
(* Funkce pro variantu ohrevu cislo 1,
kdy se voda v zasobniku ohriva pomoci topneho telesa
kdykoliv je to mozne. Dodana energie je teda primo umerna
spotrebe. Vstupem do funkce je casovy usek, kdy probiha odber,
pripadne casovy usek mezi dvema odbery a odebrany objem teple
vody v litrech. Jelikoz se cely casovy usek odberu vyhodnocuje
pouze jednou na zacatku odberu, je zde v pripade odberu delsiho
nez casova konstanta periodaSpinani zarazen while cyklus,
ktery umoznuje vyhodnocovat a pripadne i realizovat odber vzdy po
casovem useku periodaSpinani. Pokud by v prubehu casoveho useku
na vstupu zacal vysoky tarif, zavola se funkce VysokyTarifBojler[]
vstup: casovy usek jednoho odberu nebo pauzy mezi odbery,
objem odebrane vody v litrech
vystup: *)
NizkyTarifBojler[tstart_, tend_, odber_] :=
Module[{ret, trvani, pristi, cas, Q, tstarthelp, tendhelp, odber1},
If[tend - tstart > periodaSpinani, {
tstarthelp = tstart;
tendhelp = tstarthelp + periodaSpinani;
odber1 = odber;
```

```

While[tstarthelp < tend,
  {Q = c * ρ *  $\frac{\text{Pi} * \text{prumer}^2}{4}$  * (teplotaOhrivaneVody * poloha -
    NIntegrate[funkce[x], {x, 0, poloha}, AccuracyGoal → 3]);
  (* kolik tepla je potreba aktualne dodat *)

  {ret, trvani, pristi} = NizkyTarifOhrevuVody[tstarthelp];
  cas = Min[trvani, tendhelp - tstarthelp - 1,  $\frac{Q}{\text{vykonBojleru}}$ ];

  If[tstarthelp + trvani < tendhelp && tend ≠ tendhelp,
    {Ohrev[cas, tstarthelp, tstarthelp + trvani,  $\frac{\text{trvani}}{\text{tend} - \text{tstart}}$  * odber1,
      vykonBojleru * cas, 1], VysokyTarifBojler[tstarthelp + trvani, tend,
      odber1 -  $\frac{\text{tstarthelp} + \text{trvani} - \text{tstart}}{\text{tend} - \text{tstart}}$  * odber1], tstarthelp = tend},
    {Ohrev[cas, tstarthelp, tendhelp,  $\frac{\text{tendhelp} - \text{tstarthelp}}{\text{tend} - \text{tstart}}$  * odber1,
      vykonBojleru * cas, 1];
    odber1 -=  $\frac{\text{tendhelp} - \text{tstarthelp}}{\text{tend} - \text{tstart}}$  * odber1;
    If[tstarthelp + periodaSpinani ≥ tend && ret == 1,
      tstarthelp = tendhelp, tstarthelp = tstarthelp + periodaSpinani];
    If[tendhelp + periodaSpinani ≥ tend && ret == 1, tendhelp = tend,
      tendhelp = tendhelp + periodaSpinani]]]]],
  {Q = c * ρ *  $\frac{\text{Pi} * \text{prumer}^2}{4}$  * (teplotaOhrivaneVody * poloha -
    NIntegrate[funkce[x], {x, 0, poloha}, AccuracyGoal → 3]);
  (* kolik tepla je potreba aktualne dodat *)
  {ret, trvani, pristi} = NizkyTarifOhrevuVody[tstart],
  cas = Min[trvani, tend - tstart - 1,  $\frac{Q}{\text{vykonBojleru}}$ ],
  Ohrev[cas, tstart, tend, odber, vykonBojleru * cas, 1]}}];
];

```

```
In[630]:= (* Urci, kdy zacne platit nizky tarif a v danou dobu zavola
prislusnou funkci. Pokud nizky tarif nezacina primo v cas tstart,
tak se zavola funkce PrestupTepla pokryvajici casovy
interval tstart az zacatek startu nizkeho tarifu
vstup: casovy usek jednoho odberu nebo pauzy mezi odbery,
objem odebrane vody v litrech
vystup: *)
VysokyTarifBojler[tstart_, tend_, odber_] := Module[{ret, trvani, pristi},
  {ret, trvani, pristi} = NizkyTarifOhrevuVody[tstart];
  If[pristi < tend,
    {If[tstart ≠ pristi, {PrestupTepla[tstart, pristi,  $\frac{pristi - tstart}{tend - tstart} * odber$ ]}}],
    NizkyTarifBojler[pristi, tend, odber -  $\frac{pristi - tstart}{tend - tstart} * odber$ ]},
  {PrestupTepla[tstart, tend, odber]}}];
];
```

### In[631]:= **Varianta 2 – ohřev pomocí TČ kdykoliv je to možné;**

```
(* Funkce pro variantu ohrevu cislo 2,
kdy se voda v zasobniku ohřiva pomoci tepelneho cerpadla
kdykoliv je to mozne. Dodana energie je zavisla na COP cerpadla,
ktere je zavisle na aktualni venkovni teplote. Vstupem
do funkce je casovy usek, kdy probiha odber,
pripadne casovy usek mezi dvema odbery a odebrany objem teple
vody v litrech. Jelikoz se cely casovy usek odberu vyhodnocuje
pouze jednou na zacatku odberu, je zde v pripade odberu delsiho
nez casova konstanta periodaSpinani zarazen while cyklus,
ktery umoznuje vyhodnocovat a pripadne i realizovat odber vzdy po
casovem useku periodaSpinani. Pokud by v prubehu casoveho useku na
vstupu zacal vysoky tarif, zavola se funkce VysokyTarifCerpadlo[]
vstup: casovy usek jednoho odberu nebo pauzy mezi odbery,
objem odebrane vody v litrech
vystup: *)
```

```
NizkyTarifCerpadlo[tstart_, tend_, odber_] := Module[{ret, trvani, pristi,
  cas, Q, QCerpadlo, tendCerpadlo, tstarthelp, tendhelp, odber1},
  If[tend - tstart > periodaSpinani, {
    tstarthelp = tstart;
    tendhelp = tstarthelp + periodaSpinani;
    odber1 = odber;
    While[tstarthelp < tend,
      {{ret, trvani, cas} = VypocetCasu[tstarthelp, tendhelp];
      If[tstarthelp + trvani < tendhelp && tend ≠ tendhelp,
```

```

{Ohrev[cas, tstarthelp, tstarthelp + trvani,  $\frac{\text{trvani}}{\text{tend} - \text{tstart}} * \text{odber1}$ ,
  NIntegrate[TCPtep[fceTeplot[[den]][t], teplota0hrivaneVody],
  {t, tstarthelp, tstarthelp + cas}, AccuracyGoal  $\rightarrow$  0], 2],
  VysokyTarifCerpadlo[tstarthelp + trvani, tend,
  odber1 -  $\frac{\text{trvani}}{\text{tend} - \text{tstart}} * \text{odber1}$ ], tstarthelp = tend},
{Ohrev[cas, tstarthelp, tendhelp,  $\frac{\text{tendhelp} - \text{tstarthelp}}{\text{tend} - \text{tstart}} * \text{odber1}$ ,
  NIntegrate[TCPtep[fceTeplot[[den]][t], teplota0hrivaneVody],
  {t, tstarthelp, tstarthelp + cas}, AccuracyGoal  $\rightarrow$  0], 2];
odber1 -=  $\frac{\text{tendhelp} - \text{tstarthelp}}{\text{tend} - \text{tstart}} * \text{odber1}$ ;
If[tstarthelp + periodaSpinani  $\geq$  tend && ret == 1,
  tstarthelp = tendhelp, tstarthelp += periodaSpinani];
If[tendhelp + periodaSpinani  $\geq$  tend && ret == 1,
  tendhelp = tend, tendhelp += periodaSpinani];
]]] ]],
{{ret, trvani, cas} = VypocetCasu[tstart, tend];
Ohrev[cas, tstart, tend, odber,
  NIntegrate[TCPtep[fceTeplot[[den]][t], teplota0hrivaneVody],
  {t, tstart, tstart + cas}, AccuracyGoal  $\rightarrow$  0], 2];}];
];

```

In[633]:= (\* Urci, kdy zacne platit nizky tarif a v danou dobu zavola prislusnou funkci. Pokud nizky tarif nezacina primo v cas tstart, tak se zavola funkce PrestupTepla pokryvajici casovy interval tstart az zacatek startu nizkeho tarifu  
vstup: casovy usek jednoho odberu nebo pauzy mezi odbery, objem odebrane vody v litrech  
vystup: \*)

```

VysokyTarifCerpadlo[tstart_, tend_, odber_] := Module[{ret, trvani, pristi},
  {ret, trvani, pristi} = NizkyTarifTepelnehoCerpadla[tstart];
  If[pristi < tend,
    {If[tstart  $\neq$  pristi, {PrestupTepla[tstart, pristi,  $\frac{\text{pristi} - \text{tstart}}{\text{tend} - \text{tstart}} * \text{odber}$ ]}}],
    NizkyTarifCerpadlo[pristi, tend, odber -  $\frac{\text{pristi} - \text{tstart}}{\text{tend} - \text{tstart}} * \text{odber}$ ],
    {PrestupTepla[tstart, tend, odber]}}];
];

```

In[634]:=

**Varianta 3 – ohřev pomocí TČ když klesne hladina TV pod danou úroveň;**

(\* Funkce pro variantu ohrevu cislo 3,  
 kdy se voda v zasobniku ohriava pomoci tepelneho cerpadla v pripade,  
 kdy objem teple vody klesne pod urcenou hodnotu OffsetDown. Dodana  
 energie je zavisla na COP cerpadla, ktere je zavisle na aktualni  
 venkovni teplote. Vstupem do funkce je casovy usek, kdy probiha odber,  
 pripadne casovy usek mezi dvema odbery a objem odebrane teple  
 vody v litrech. Na zacatku funkce se spocita casOdkladu,  
 který rika za jak dlouho bude s aktualnim odberem objem teple vody na  
 hranicni hodnotu. Pokud tento okamzik nastane za dobu aktualniho odberu,  
 spusti se ohrev. Pokud by aktualni odber skoncil drive,  
 nez bude ohraty cely objem vody, ohrev bude pokracovat i v dalsim odberu/pauze  
 mezi odbery. Pokud by v prubehu casoveho useku na vstupu zacal vysoky tarif,  
 zavola se funkce VysokyTarifCerpadlo[]  
 vstup: casovy usek jednoho odberu nebo pauzy mezi odbery,  
 objem odebrane vody v litrech  
 vystup: \*)

```
NizkyTarifCerpadloOffset[tstart_, tend_, odber_] :=
Module[{casOdkladu, ret, trvani, pristi, cas, ohrev, bilance, Q,
  QCerpadlo, tendCerpadlo, tstart1, odber1, rozhrani, interval},
  tstart1 = tstart;
  odber1 = odber;
  If[odber1 > 0, casOdkladu =  $\frac{1000 * \text{Pi} * \text{prumer}^2}{4 * \text{odber1}} *
    (\text{OffsetDown} - \text{poloha}) * (\text{tend} - \text{tstart1})$ , casOdkladu = 0];
  If[casOdkladu < tend - tstart1 && casOdkladu > 0 && tstart != tend,
    {PrestupTepla[tstart1, tstart1 + casOdkladu,  $\frac{\text{casOdkladu}}{\text{tend} - \text{tstart}} * \text{odber1}$ ],
      tstart1 = tstart1 + casOdkladu, odber1 -=  $\frac{\text{casOdkladu}}{\text{tend} - \text{tstart}} * \text{odber1}$ },
    {If[casOdkladu > tend - tstart1 && tstart1 != tend, {PrestupTepla[tstart1,
      tend, odber1], tstart1 = tend, odber1 = 0}], casOdkladu = 0}];

  {ret, trvani, cas} = VypocetCasu[tstart1, tend];
  If[ $\left( \text{poloha} + \frac{4 * \text{odber1}}{10^3 * \text{Pi} * \text{prumer}^2} > \text{OffsetDown} \right) \vee \left( \text{poloha} > \text{OffsetDown} \vee \right) \vee$ 
     $\left( \text{ohratCele} == 1 \right) \&\& \text{tstart1} + \text{cas} < \text{tend} \&\& \text{cas} > 60 \&\& \text{tstart1} \neq \text{tend}$ ,
    {If[ $\text{poloha} + \frac{4 * \text{odber1}}{10^3 * \text{Pi} * \text{prumer}^2} > \text{OffsetDown}$ , ohratCele = 1]};
  Q = NIntegrate[TCPtep[fceTeplo[t][den]][t], teplotaOhrivaneVody],
    {t, tstart1, tstart1 + cas}, AccuracyGoal -> 0];
  (* teplo, ktere bude cerpadlem dodano za dany cas *)
  interval = 0.01;
  rozhrani = 0;
  While[interval >= 10^-5,
```



```

While[ Q - c * ρ *  $\frac{\text{Pi} * \text{prumer}^2}{4}$  *
      (teplotaOhrivaneVody * (rozhrani + interval) - NIntegrate[funkce[x],
        {x, 0, (rozhrani + interval)}, AccuracyGoal → 3]) > 10,
      {rozhrani += interval}
];
interval /= 10;
];
ohrev = objem * rozhrani;
dodanaEnergie =
Append[dodanaEnergie, {tstart1 + cas, NIntegrate[TCPel[fceTeplot[[den]][t],
  teplotaOhrivaneVody], {t, tstart1, tstart1 + cas}, AccuracyGoal → 0]}}];
PrestupTepla[tstart1, tstart1 + cas, -ohrev +  $\frac{\text{cas}}{\text{tend} - \text{tstart1}}$  * odber1],
odber1 -=  $\frac{\text{cas}}{\text{tend} - \text{tstart1}}$  * odber1, tstart1 = tstart1 + cas;
}, cas = 0];
{ret, trvani, cas} = VypocetCasu[tstart1, tend];
If[poloha < 0.05, ohratCele = 0];
If[tend > tstart1 + cas + 1 && cas > 0 && odber1 > 0 && tstart1 ≠ tend,
  {NizkyTarifCerpadloOffset[tstart1 + cas, tend,
    odber1 -  $\frac{\text{cas}}{\text{tend} - \text{tstart1}}$  * odber1]}, {{ret, trvani, pristi} =
    NizkyTarifTepelnehoCerpadla[tstart1 + cas], If[ret == 1 && tstart1 ≠ tend,
      {PrestupTepla[tstart1, tend,  $\frac{\text{cas}}{\text{tend} - \text{tstart1}}$  * odber1]},
      If[tstart1 ≠ tend && tstart1 + cas < tend, VysokyTarifCerpadloOffset[
        tstart1 + cas, tend, odber1 -  $\frac{\text{cas}}{\text{tend} - \text{tstart1}}$  * odber1]]]]];
];

```

```

In[636]:= (* Urci, kdy zacne platit nizky tarif a v danou dobu zavola
prislusnou funkci. Pokud nizky tarif nezacina primo v cas tstart,
tak se zavola funkce PrestupTepla pokryvajici casovy
interval tstart az zacatek startu nizkeho tarifu
vstup: casovy usek jednoho odberu nebo pauzy mezi odbery,
objem odebrane vody v litrech
vystup: *)
VysokyTarifCerpadloOffset[tstart_, tend_, odber_] :=
Module[{ret, trvani, pristi},
{ret, trvani, pristi} = NizkyTarifTepelnehoCerpadla[tstart];
If[pristi < tend && tstart ≠ tend,
{If[tstart ≠ pristi, PrestupTepla[tstart, pristi,
 $\frac{\text{pristi} - \text{tstart}}{\text{tend} - \text{tstart}} * \text{odber}$ ]}],
NizkyTarifCerpadloOffset[pristi, tend, odber -
 $\frac{\text{pristi} - \text{tstart}}{\text{tend} - \text{tstart}} * \text{odber}$ ]}],
{PrestupTepla[tstart, tend, odber]}}];
];

```

In[637]:=

#### Varianta 4 – ohřev pomocí TČ optimalizováno dle teploty;

```

(* Funkce pro variantu ohrevu cislo 4,
kdy se voda v zasobniku ohřiva pomoci tepelneho cerpadla v pripade,
kdy objem teple vody klesne pod urcenou hodnotu OffsetDown nebo
je idealni casovy usek pro ohřev vody optimalizovany dle venkovni
teploty. Dodana energie je zavisla na COP cerpadla, ktere je zavisle
na aktualni venkovni teplotě. Vstupem do funkce je casovy usek,
kdy probiha odber, pripadne casovy usek mezi dvema odbery a objem
odebrane teple vody v litrech. Na zacatku funkce se spocita casOdkladu,
ktery rika za jak dlouho bude s aktualnim odberem objem teple vody
na hranicni hodnote nebo za jak dlouho bude idealni casovy usek pro
ohřev vody. Pokud tento okamzik nastane za dobu aktualniho odberu,
spusti se ohřev. Pokud by aktualni odber skoncil drive,
nez bude ohraty cely objem vody,
ohřev bude pokracovat i v dalsim odberu/pauze mezi odbery. Pokud
se ohřev spusti z duvody prislis nizkeho objemu teple vody,
ohřev bude trvat jen kratkou chvili (15 min), aby voda nedosla,
ale zaroven se nebude ohřivat cely zasobnik mimo idealni cas
ohřevu. Pokud by v prubehu casoveho useku na vstupu zacal vysoky tarif,
zavola se funkce VysokyTarifCerpadlo[]
vstup: casovy usek jednoho odberu nebo pauzy mezi odbery,
objem odebrane vody v litrech
vystup: *)
NizkyTarifCerpadloOptimalizace[tstart_, tend_, odber_] :=
Module[{casOdkladu, casOdkladu1, casOdkladu2, ret,
trvani, pristi, cas, ohřev, bilance, Q, QCerpadlo, tendCerpadlo,

```

```

tstart1, odber1, cas0hrevu, index, help, interval, rozhrani},
tstart1 = tstart;
odber1 = odber;
index = 1;
help = 1;
While[casy0hrevu[[index, 2]] < tstart1 && help == 1,
  If[index + 1 ≤ Length[casy0hrevu], index += 1, help = 0];
];
cas0hrevu = casy0hrevu[[index]];
cas0dkladu1 = cas0hrevu[[1]] - tstart1;
If[odber1 > 0, cas0dkladu2 =  $\frac{1}{4 * odber1} * 1000 * (OffsetDown - poloha) * Pi * prumer^2 * (tend - tstart1)$ , cas0dkladu2 = 0];
If[(cas0dkladu1 ≤ cas0dkladu2 || cas0dkladu2 == 0) && cas0dkladu1 > 0,
  {cas0dkladu = cas0dkladu1, ohratCele = 1}, cas0dkladu = cas0dkladu2];
If[cas0dkladu < tend - tstart1 && cas0dkladu > 0 && tstart ≠ tend,
  {PrestupTepla[tstart1, tstart1 + cas0dkladu,  $\frac{cas0dkladu}{tend - tstart} * odber1$ ],
  tstart1 = tstart1 + cas0dkladu, odber1 = odber1 -  $\frac{cas0dkladu}{tend - tstart} * odber1$ },
  {If[cas0dkladu > tend - tstart1 && tstart1 ≠ tend, {{ret, trvani, cas} =
  VypocetCasu[tstart1, tend], If[tstart1 > cas0hrevu[[1]] &&
  tstart1 < cas0hrevu[[1]] + cas, {PrestupTepla[tstart1, tend,
  odber1 - NIntegrate[TCPtep[fceTeplot[[den]][t], teplotaOhrivaneVody],
  {t, tstart1, tstart1 + cas}, AccuracyGoal → 0]},
  dodanaEnergie = Append[dodanaEnergie, {tstart1 + cas,
  NIntegrate[TCPel[fceTeplot[[den]][t], teplotaOhrivaneVody],
  {t, tstart1, tstart1 + cas}, AccuracyGoal → 0]}]},
  {PrestupTepla[tstart1, tend, odber1]}], tstart1 =
  tend, 0}], cas0dkladu = 0}];
{ret, trvani, cas} = VypocetCasu[tstart1, tend];
If[cas0hrevu[[1]] > tstart1 && cas > 900, {cas = 900}];
If[ $\left( poloha + \frac{4 * odber1}{10^3 * Pi * prumer^2} > OffsetDown || poloha > OffsetDown || \right)$ 
  ohratCele == 1] && tstart1 + cas < tend && cas > 60 && tstart1 ≠ tend,
  {bilance[rozhrani_?NumericQ] :=  $c * \rho * \frac{Pi * prumer^2}{4} * (teplotaOhrivaneVody * rozhrani - NIntegrate[funkce[x], {x, 0, rozhrani}], AccuracyGoal → 3) - NIntegrate[TCPtep[fceTeplot[[den]][t], teplotaOhrivaneVody], {t, tstart1, tstart1 + cas}, AccuracyGoal → 0];$ 
  ohrev = objem * FindRoot[bilance[rozhrani] == 0, {rozhrani,
  NIntegrate[TCPtep[fceTeplot[[den]][t], teplotaOhrivaneVody],
  {t, tstart1, tstart1 + cas}, AccuracyGoal → 0] /
  (objem * ρ * c * (teplotaOhrivaneVody - teplotaStudeneVody))},
  AccuracyGoal → 3][[1, 2]];

```

```

PrestupTepla[tstart1, tstart1 + cas, -ohrev +  $\frac{\text{cas}}{\text{tend} - \text{tstart1}} * \text{odber1}$ ],
dodanaEnergie =
  Append[dodanaEnergie, {tstart1 + cas, NIntegrate[TCPel[fceTeplot[[den]][t],
    teplotaOhrivaneVody], {t, tstart1, tstart1 + cas}, AccuracyGoal -> 0]}],
  odber1 = odber1 -  $\frac{\text{cas}}{\text{tend} - \text{tstart1}} * \text{odber1}$ , tstart1 = tstart1 + cas;
}, cas = 0];
{ret, trvani, cas} = VypocetCasu[tstart1, tend];
If[casOhrevu[[1]] > tstart1 && casOhrevu[[1]] < tend && cas > 900, {cas = 900}];
If[poloha < 0.05 || tstart1 < casOhrevu[[1]] ||
  tstart1 > casOhrevu[[2]], ohratCele = 0];
If[tend > tstart1 + cas + 1 && cas > 0 && odber1 > 0 && tstart1 ≠ tend,
  {NizkyTarifCerpadloOptimalizace[tstart1 + cas, tend,
    odber1 -  $\frac{\text{cas}}{\text{tend} - \text{tstart1}} * \text{odber1}$ ]}], {{ret, trvani, pristi} =
  NizkyTarifTepelnehoCerpadla[tstart1 + cas], If[ret == 1 && tstart1 ≠ tend,
  PrestupTepla[tstart1, tend,  $\frac{\text{cas}}{\text{tend} - \text{tstart1}} * \text{odber1}$ ],
  If[tstart1 ≠ tend && tstart1 + cas < tend, VysokyTarifCerpadloOptimalizace[
    tstart1 + cas, tend, odber1 -  $\frac{\text{cas}}{\text{tend} - \text{tstart1}} * \text{odber1}$ ]]]]];
];

```

In[639]= (\* Urci, kdy zacne platit nizky tarif a v danou dobu zavola prislusnou funkci. Pokud nizky tarif nezacina primo v cas tstart, tak se zavola funkce PrestupTepla pokryvajici casovy interval tstart az zacatek startu nizkeho tarifu  
vstup: casovy usek jednoho odberu nebo pauzy mezi odbery, objem odebrane vody v litrech  
vystup: \*)

```

VysokyTarifCerpadloOptimalizace[tstart_, tend_, odber_] :=
  Module[{ret, trvani, pristi},
    {ret, trvani, pristi} = NizkyTarifTepelnehoCerpadla[tstart];
    If[pristi < tend,
      {If[tstart ≠ pristi, PrestupTepla[tstart, pristi,  $\frac{\text{pristi} - \text{tstart}}{\text{tend} - \text{tstart}} * \text{odber}$ ]],
      NizkyTarifCerpadloOptimalizace[pristi, tend,
        odber -  $\frac{\text{pristi} - \text{tstart}}{\text{tend} - \text{tstart}} * \text{odber}$ ]}], {PrestupTepla[tstart, tend, odber]};
  ];

```

```

In[640]:= (* Vypocita, kolik litru vody se za dany cas ohreje a zavola funkci
PrestupTepla. Dale pocita spotrebovanou elektrickou energii pri ohrevu vody.
vstup: cas ohrevu, start casoveho useku, konec casoveho useku,
odber behem casoveho useku, dodany vykon pri ohrevu teple vody za cas,
zpusob ohrevu (1 = elektrika, 2 = TC)
vystup: *)
Ohrev[cas_, tstart_, tend_, odber_, dodanyVykon_, zpusobOhrevu_] :=
Module[{ohrev, bilance, cas1, interval, rozhrani},
cas1 = cas;
If[cas1 > 60, {
interval = 0.01;
rozhrani = 0;
While[interval >= 10^-5 && rozhrani <= 1,
While[dodanyVykon - c * rho *  $\frac{\text{Pi} * \text{prumer}^2}{4} * (\text{teplotaOhrivaneVody} *
(\text{rozhrani} + \text{interval}) - \text{NIntegrate}[\text{funkce}[x], \{x, 0,
(\text{rozhrani} + \text{interval})\}, \text{AccuracyGoal} \rightarrow 3]) > 10 \&\& \text{rozhrani} \leq 1,$ 
{rozhrani += interval}
];
interval /= 10;
];
ohrev = objem * rozhrani;
PrestupTepla[tstart, tstart + cas1, -ohrev +  $\frac{\text{cas1}}{\text{tend} - \text{tstart}} * \text{odber}$ ];
PrestupTepla[tstart + cas1, tend, odber -  $\frac{\text{cas1}}{\text{tend} - \text{tstart}} * \text{odber}$ ];
If[zpusobOhrevu == 1,
dodanaEnergie = Append[dodanaEnergie, {tstart + cas1, dodanyVykon}],
dodanaEnergie = Append[dodanaEnergie, {tstart + cas1,
NIntegrate[TCPel[fceTeplot[[den]][t], teplotaOhrivaneVody],
{t, tstart, tstart + cas1}, AccuracyGoal -> 0]}]},
{cas1 = 0, PrestupTepla[tstart, tend, odber]}];
];

```

```

In[641]:= (* Vypocita, kolik je potreba aktualne dodat tepla pro ohrati veskere vody v
zasobniku a jak dlouho by to trvalo. Pokud je tato hodnota mensi nez doba
trvani aktualniho odberu/pauzy mezi odbery a dobou platnosti nizkeho tarifu,
ohreje se cely zasobnik. V opacnem pripade ohrev trva jen do konce
platnosti nizkeho tarifu nebo konce odbery/pauzy mezi odbery.
vstup: casovy usek jednoho odberu nebo pauzy mezi odbery
vystup: logicka hodnota o platnosti nizkeho tarifu (0= ne, 1= ano),
jak dlouho bude jeste nizky tarif trvat, vysledny cas ohrevu *)
VypocetCasu[tstart_, tend_] :=
Module[{tendCerpadlo, QCerpadlo, Q, cas, ret, trvani, pristi},
  Q = c * rho *  $\frac{\text{Pi} * \text{prumer}^2}{4} * (\text{teplotaOhrivaneVody} * \text{poloha} -$ 
    NIntegrate[funkce[x], {x, 0, poloha}, AccuracyGoal -> 3]);
  (* kolik tepla je potreba aktualne dodat *)
  interval = 100;
  tendCerpadlo = tstart;
  While[interval > 10^-2,
    While[Q - NIntegrate[TCPTep[fceTeplo[[den]][t], teplotaOhrivaneVody],
      {t, tstart, tendCerpadlo + interval}, AccuracyGoal -> 3] > 5,
      {tendCerpadlo += interval}
    ];
    interval /= 10;
  ];
  {ret, trvani, pristi} = NizkyTarifTepelnehoCerpadla[tstart];
  cas = Min[trvani, tend - tstart - 1, tendCerpadlo - tstart];
  {ret, trvani, cas}];

```

In[642]= **Funkce pro odběr a ohřev vody;**

```
(* na zaklade varianty ohrevu zavola prislusnou
funkci pro ohrev (NizkyTarifxxx nebo VysokyTarifxxx)
pro samodny odber a pote i pro pauzu mezi dvema odbery
vstup: cas a konec jednoho odberu, objem odebrane teple vody,
cas a konec jedne pauzy mezi odbery, varianta ohrev (1 - 4)
vystup: *)
Ohrivej[tstartOdber_, tendOdber_, objemOdber_, tstartOhrev_,
tendOhrev_, varianta_] := Module[{Q, ret, trvani, pristi},
If[varianta == 1, {{ret, trvani, pristi} = NizkyTarifOhrevuVody[tstartOdber];
If[ret != 0, NizkyTarifBojler[tstartOdber, tendOdber, objemOdber],
VysokyTarifBojler[tstartOdber, tendOdber, objemOdber]],
{ret, trvani, pristi} = NizkyTarifOhrevuVody[tstartOhrev];
If[ret != 0, NizkyTarifBojler[tstartOhrev, tendOhrev, 0],
VysokyTarifBojler[tstartOhrev, tendOhrev, 0]];]];

If[varianta == 2,
{{ret, trvani, pristi} = NizkyTarifTepelnehoCerpada[tstartOdber];
If[ret != 0, NizkyTarifCerpadlo[tstartOdber, tendOdber, objemOdber],
VysokyTarifCerpadlo[tstartOdber, tendOdber, objemOdber]],
{ret, trvani, pristi} = NizkyTarifTepelnehoCerpada[tstartOhrev];
If[ret != 0, NizkyTarifCerpadlo[tstartOhrev, tendOhrev, 0],
VysokyTarifCerpadlo[tstartOhrev, tendOhrev, 0]];]];

If[varianta == 3,
{{ret, trvani, pristi} = NizkyTarifTepelnehoCerpada[tstartOdber];
If[ret != 0, NizkyTarifCerpadloOffset[tstartOdber, tendOdber, objemOdber],
VysokyTarifCerpadloOffset[tstartOdber, tendOdber, objemOdber]],
{ret, trvani, pristi} = NizkyTarifTepelnehoCerpada[tstartOhrev];
If[ret != 0, NizkyTarifCerpadloOffset[tstartOhrev, tendOhrev, 0],
VysokyTarifCerpadloOffset[tstartOhrev, tendOhrev, 0]];]];

If[varianta == 4,
{{ret, trvani, pristi} = NizkyTarifTepelnehoCerpada[tstartOdber];
If[ret != 0, NizkyTarifCerpadloOptimalizace[
tstartOdber, tendOdber, objemOdber],
VysokyTarifCerpadloOptimalizace[tstartOdber, tendOdber, objemOdber]],
{ret, trvani, pristi} = NizkyTarifTepelnehoCerpada[tstartOhrev];
If[ret != 0, NizkyTarifCerpadloOptimalizace[tstartOhrev, tendOhrev, 0],
VysokyTarifCerpadloOptimalizace[tstartOhrev, tendOhrev, 0]];]];
];
```

```

In[644]:= (* nacte odbery a vola funkce pro ohrev vody,
tvori funkce o stavu teple vody v zasobniku v case
vstup: varianta ohrevu
vystup: *)
NactiOdbery[varianta_] := Module[{fun, ret, trvani, pristi, Q, cas, ohrev},
  If[varianta == 4, {VypocetCasuOptimalizace[], OffsetDown = 0.95;};
  Ohrivej[0, 1, 0, 1, odbery[[2, 1]], varianta];
  Ohrivej[odbery[[#, 1]], odbery[[#, 2]], odbery[[#, 3]], odbery[[#, 2]],
    odbery[[# + 1, 1]], varianta] & /@ Range[2, Length[odbery] - 1];
  Ohrivej[odbery[[Length[odbery], 1]], odbery[[Length[odbery], 2]], odbery[[
    Length[odbery], 3]], odbery[[Length[odbery], 2]], 24 * 3600, varianta];

  {odberyVCase = Append[odberyVCase, {odbery[[#, 1]] - 1, 0}];
  odberyVCase =
    Append[odberyVCase, {odbery[[#, 1]],  $\frac{\text{odbery}[[\#, 3]]}{\text{odbery}[[\#, 2]] - \text{odbery}[[\#, 1]]}$ }]};
  odberyVCase = Append[odberyVCase, {odbery[[#, 2]],
     $\frac{\text{odbery}[[\#, 3]]}{\text{odbery}[[\#, 2]] - \text{odbery}[[\#, 1]]}$ }]};
  odberyVCase = Append[odberyVCase, {odbery[[#, 2]] + 1, 0}];} & /@
  Range[2, Length[odbery]];
  odberyVCase = Append[odberyVCase, {24 * 3600, 0}];

  fceOdbery = Interpolation[odberyVCase, InterpolationOrder → 1];
  fceTeplaVoda = Interpolation[objemTepleVodyVCase, InterpolationOrder → 1];
  fceDodatTepla = Interpolation[dodatTeplaVCase, InterpolationOrder → 1];
];

```



In[645]:= **Funkce udávající platnost jednotlivých distribučních tarifů elektřiny;**

(\* rika, zda je aktualne platny nizky tarif  
distribucni sazby elektriny pro tepelna cerpadla  
vstup: cas startu tepelneho cerpadla  
vystup: logicka hodnota zda je aktualne platny nizky tarif (1- ano, 0- ne),  
jak dlouho jeste bude nizky tarif trvat,  
pokud neni platny, tak za jak dlouho bude platny \*)

```
NizkyTarifTepelnehoCerpada[tstart_] := Module[{ret, pristi, trvani},
  ret = 0;
  pristi = 0;
  trvani = 0;
  If[0 ≤ tstart < 6 * 3600, {ret = 1, trvani = 6 * 3600 - tstart}];
  If[6 * 3600 ≤ tstart < 7 * 3600, pristi = 7 * 3600];
  If[7 * 3600 ≤ tstart < 9 * 3600, {ret = 1, trvani = 9 * 3600 - tstart}];
  If[9 * 3600 ≤ tstart < 10 * 3600, pristi = 10 * 3600];
  If[10 * 3600 ≤ tstart < 13 * 3600, {ret = 1, trvani = 13 * 3600 - tstart}];
  If[13 * 3600 ≤ tstart < 14 * 3600, pristi = 14 * 3600];
  If[14 * 3600 ≤ tstart < 16 * 3600, {ret = 1, trvani = 16 * 3600 - tstart}];
  If[16 * 3600 ≤ tstart < 17 * 3600, pristi = 17 * 3600];
  If[17 * 3600 ≤ tstart < 24 * 3600, {ret = 1, trvani = 24 * 3600 - tstart}];
  {ret, trvani, pristi}];
```

In[647]:= (\* rika, zda je aktualne platny nizky  
tarif distribucni sazby elektriny pro bojleru  
vstup: cas startu bojleru  
vystup: logicka hodnota zda je aktualne platny nizky tarif (1- ano, 0- ne),  
jak dlouho jeste bude nizky tarif trvat,  
pokud neni platny, tak za jak dlouho bude platny \*)

```
NizkyTarifOhrevuVody[tstart_] := Module[{ret, pristi, trvani},
  ret = 0;
  pristi = 0;
  trvani = 0;
  If[0 ≤ tstart < 6 * 3600, {ret = 1, trvani = 6 * 3600 - tstart}];
  If[6 * 3600 ≤ tstart < 19 * 3600, pristi = 19 * 3600];
  If[19 * 3600 ≤ tstart < 21 * 3600, {ret = 1, trvani = 21 * 3600 - tstart}];
  If[21 * 3600 ≤ tstart < 24 * 3600, pristi = 24 * 3600];
  {ret, trvani, pristi}];
```

In[648]= **Načtení výkonu tepelného čerpadla a denních teplot;**

```
(* nacte vykonove parametry tepelneho cerpadla,
vytvori funkce pro prikony, vykon a COP
vstup:
vystup: *)
NactiTepelneCerpadlo[] :=
Module[{T, n, Ptep45, Ptep55, Ptep60, Pel45, Pel55, Pel60, radekEl45, radekEl55,
radekEl60, radekTep45, radekTep55, radekTep60, dataEl, dataTep},
(* TC vzduch-voda Vitocal 222-A *)
(* https://www.viessmann.cz/cs/obytne-budovy/tepelna-
cerpadla/tepelna-cerpadla-vzduchvoda/vitocal-222a.html *)
T = {-20, -15, -7, 2, 7, 10, 20, 30};
n = Length[T];
Ptep45 = {3.88, 4.38, 5.41, 5.43, 5.06, 6.65, 7.85, 8.93};
Ptep55 = {0, 4.47, 5.56, 2.88, 4.4, 4.53, 5.58, 6.95};
Ptep60 = {0, 0, 5.12, 3.01, 4.14, 4.44, 5.48, 6.81};
Pel45 = {2.21, 2.29, 2.4, 1.06, 1.28, 1.26, 1.25, 1.22};
Pel55 = {10^-6, 2.67, 2.84, 1.33, 1.51, 1.57, 1.57, 1.55};
Pel60 = {10^-6, 10^-6, 2.89, 1.49, 1.75, 1.76, 1.77, 1.76};

(*vytvoreni listu dat s hodnotami*)
radekEl45[i_] := {T[[i]], 45., Pel45[[i]]};
radekEl55[i_] := {T[[i]], 55., Pel55[[i]]};
radekEl60[i_] := {T[[i]], 60., Pel60[[i]]};
radekTep45[i_] := {T[[i]], 45., Ptep45[[i]]};
radekTep55[i_] := {T[[i]], 55., Ptep55[[i]]};
radekTep60[i_] := {T[[i]], 60., Ptep60[[i]]};
dataEl =
Flatten[Table[{radekEl45[i], radekEl55[i], radekEl60[i]}, {i, n}], 1];
dataTep = Flatten[Table[{radekTep45[i], radekTep55[i],
radekTep60[i]}, {i, n}], 1];

(*vyjadreni prubehu zavislosti energetickych hodnot na Tvzduch, Tvoda*)
(*Elektricky prikony*)
TCPel[Tvzduch_, Tvoda_] =
10^3 * Interpolation[dataEl, InterpolationOrder -> {1, 1}][Tvzduch, Tvoda];
(*Topny vykon*)
TCPtep[Tvzduch_, Tvoda_] =
10^3 * Interpolation[dataTep, InterpolationOrder -> {1, 1}][Tvzduch, Tvoda];
(*Topny faktor*)
TCCOP[Tvzduch_, Tvoda_] =  $\frac{TCPtep[Tvzduch, Tvoda]}{TCPel[Tvzduch, Tvoda]}$ ;
];
```

```
In[650]:= (* nacte hodinove teploty v dany den roku 2017 a vytvori z nich casove funkce
vstup:
vystup: *)
NactiVenkovniTeploty[] := Module[{T, index},
  T = 3600 * Range[0, 24];
  teplotyPole =
    teploty[[], Range[2, Length[teploty[[]]]]] & /@ Range[1, Length[teploty]];
  index = 1;
  While[index ≤ Length[teplotyPole],
    fceTeplot = Append[fceTeplot, Interpolation[
      {T[[]], teplotyPole[index, #]} & /@ Range[1, Length[T]]];
    index += 1;
  ];
];
```

In[651]:= **Funkce pro výpočet ideálního času ohřevu ;**

```
In[652]:= (* funkce podobna funkci FindRoot, jen spolehlivejsi a rychlejsi
vstup: hodnotici funkce,
maximum v okoli ktereho si hleda dany casovy interval ohrevu,
minimalni hodnota hodnotici funkce,
ktera tvori okraje hledaneho casoveho intervalu
vystup: start a konec nalezeneho casoveho intervalu ohrevu *)
NajdiOhrevStartaOhrevEndVlastni[hodnoticiFunkce_, maximum_, hranice_] :=
  Module[{ohrevStart, ohrevEnd, interval},
    interval = 100;
    ohrevStart = maximum[[]];
    ohrevEnd = maximum[[]];
    While[interval >= 1,
      While[
        hodnoticiFunkce[ohrevStart - interval] > hranice && ohrevStart - interval ≥ 0,
        ohrevStart -= interval;
      ];
      interval /= 10;
    ];
    interval = 100;
    While[interval ≥ 1,
      While[
        hodnoticiFunkce[ohrevEnd + interval] > hranice && ohrevEnd + interval ≤ 86 400,
        ohrevEnd += interval;
      ];
      interval /= 10;
    ];
    {ohrevStart, ohrevEnd}];
```

```

In[653]:= (* vypocita idealni casy ohrevu dle algoritmu popsaneho v praci
vstup:
vystup: *)
VypocetCasuOptimalizace[] :=
Module[{min, hodnoticiFunkceBody, hodnoticiFunkce, ret, trvani, pristi, tstart,
tstartfinal, tendfinal, VTtstart, VTtend, lokalniMaximaNeserazene,
lokalniMaxima, odberyCelkem, Q, ohrevStart, ohrevEnd,
ohrev, hodnoceniList, hodnoceni, QCerpadlo},
hodnoticiFunkceBody = (10 * TCCOP[fceTeplot[[den]][#], teplotaOhrivaneVody]) ^
2 & /@ Range[0, 24 * 3600 - 1];
min = Min[hodnoticiFunkceBody];
hodnoticiFunkceBody -= 0.9 * min;
tstart = 0;
odberyCelkem = 0;
While[tstart < 24 * 3600,
{ret, trvani, pristi} = NizkyTarifTepelnehoCerpadla[tstart];
If[ret == 1, {VTtstart = tstart + trvani, tstart = VTtstart},
{VTtend = pristi, tstart = VTtend}];
If[VTtstart < VTtend, hodnoticiFunkceBody[[#]] = 0; & /@
Range[VTtstart, VTtend]];
];
lokalniMaximaNeserazene = FindPeaks[hodnoticiFunkceBody];
lokalniMaxima = Sort[lokalniMaximaNeserazene, #1[[2]] > #2[[2]] &];

If[odbery[[#, 2]] <= lokalniMaxima[[1, 1]], odberyCelkem += odbery[[#, 3]] & /@
Range[Length[odbery]]];
If[odbery[[#, 1]] < lokalniMaxima[[1, 1]] &&
odbery[[#, 2]] > lokalniMaxima[[1, 1]],
odberyCelkem +=  $\frac{\text{lokalniMaxima}[[1, 1]] - \text{odbery}[[\#, 1]]}{\text{odbery}[[\#, 2]] - \text{odbery}[[\#, 1]]} * \text{odbery}[[\#, 3]]$  & /@
Range[Length[odbery]]];
If[odberyCelkem < objemTeple * OffsetDown, odberyCelkem += 200 - objemTeple,
odberyCelkem = 200 * OffsetDown];
Q = 10^(-3) * rho * odberyCelkem * c * (teplotaOhrivaneVody - teplotaStudeneVody);
hodnoticiFunkce = Interpolation[hodnoticiFunkceBody];
casyOhrevu = NajdiIntervaly[hodnoticiFunkce, Q, lokalniMaxima];
casyOhrevu = SortBy[casyOhrevu, Last];
Print["casyOhrevu ", casyOhrevu];
];

```

```

In[654]:= (* hleda casove intervaly ohrevu takove, aby soucet dodaneho
            tepelneho vykonu v danym intervalech byl roven pozadovanemu teplu Q
            vstup: hodnotici funkce, pozadovane teplo v J,
            seznam lokalnich maxim hodnotici funkce
            vystup: seznam intervalu casu ohrevu *)
NajdiIntervaly[hodnoticiFunkce_, Q_, lokalniMaxima_] :=
Module[{hranice, interval, casy, vystup},
  hranice = lokalniMaxima[[1]];
  interval = 10;
  vystup = {};
  While[interval >= 10^-3,
    While[Qdodane[hranice[[2]] - interval,
      hodnoticiFunkce, lokalniMaxima][[1]] < Q * 1.05,
      vystup = Qdodane[hranice[[2]], hodnoticiFunkce, lokalniMaxima];
      hranice -= interval;
    ];
    interval /= 10;
  ];
  vystup[[2]]];

In[655]:= (* pocita teplo, ktere by se mohlo dodat v casovych intervalech,
            kde ma hodnotici funkce vyssi hodnotu nez hranice
            vstup:
            hranicni hodnota hodnotici funkce nutne k zapnuti/vypnuti tepelneho cerpadla
            vystup: dodane teplo, seznam intervalu casu ohrevu *)
Qdodane[hranice_, hodnoticiFunkce_, lokalniMaxima_] :=
Module[{Q, casy, maxima},
  casy = {};
  maxima = {};
  Q = 0;
  If[hranice < lokalniMaxima[[#, 2]], maxima =
    Append[maxima, lokalniMaxima[[#]]] & /@ Range[Length[lokalniMaxima]];
  casy = Append[casy, NajdiOhrevStartaOhrevEndVlastni[hodnoticiFunkce,
    maxima[[#]], hranice] & /@ Range[Length[maxima]]];
  casy = casy[[1]];
  Q += NIntegrate[TCPTep[fceTep[den]][t], teplotaOhrivaneVody], {t,
    casy[[#, 1]], casy[[#, 2]]}, AccuracyGoal -> 0] & /@ Range[Length[casy]];
  {Q,
  casy}];

```

```

In[656]:= (* simuluje funkci zasobniku teple vody za jeden den
vstup: varianta ohrevu
vystup: *)
SimulujDen[varianta_] := Module[{celkovaDodanaEnergie},
  Print["Prave bezi den ", den];
  objemTepleVodyVCase = {{0, objemTeple}};
  dodatTeplaVCase = {{0, dodatTepla}};
  odberyVCase = {{0, 0}};
  dodanaEnergie = {{0, 0}};
  casyOhrevu = {};
  NactiOdbery[varianta];
  celkovaDodanaEnergie =
    
$$\frac{1}{3.6 * 10^6} * \text{Total}[\text{dodanaEnergie}[\#, 2]] \& /@ \text{Range}[\text{Length}[\text{dodanaEnergie}]]];
  Print[ZobrazfceTeplaVoda[]];
  Print[ZobrazfceOdbery[]];

  Print["Celková dodaná energie za ",
    den, ". den je ", celkovaDodanaEnergie, " kWh."];
  Print[""];
  den = den + 1;
  dodanaEnergieDen = Append[dodanaEnergieDen, celkovaDodanaEnergie];
  Qtotal += celkovaDodanaEnergie;
];$$

```

In[657]:=

In[658]:=

### Ovládání celého programu ;

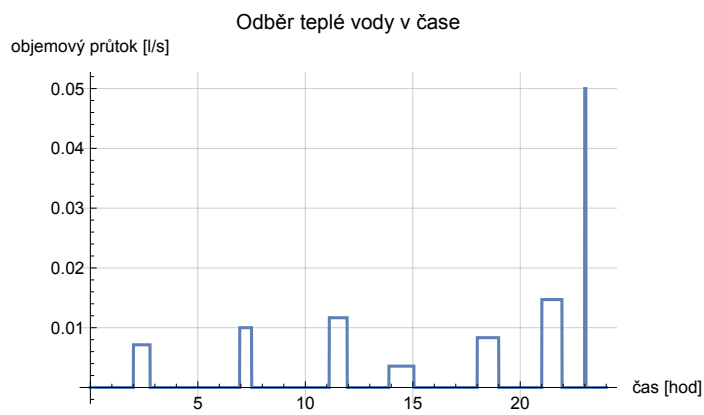
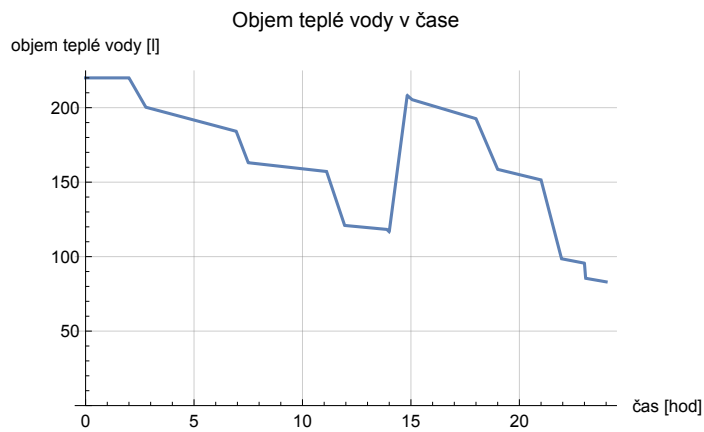
```

NactiTepelneCerpadlo[]
NactiVenkovniTeploty[]
For[i=0, i<1, i++, {SimulujDen[4]}];
Print[dodanaEnergieDen];
Print[Qtotal];

```

Prave bezi den 1

casy0hrevu {{50 401, 52 816}}



Celková dodaná energie za 1. den je 1.74903 kWh.

{1.74903}

1.74903