

ENGN2226 Portfolio:
Analysis of household hot water systems

Bennett Schneider
u5181300

October 22, 2014

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Approach	1
2	System Requirements: Human Factors	2
2.1	Waiting time	2
2.2	Temperature	3
3	System types: Preliminary time and energy analysis	3
3.1	Continuous gas	3
3.2	Instant (continuous) electric	5
3.3	Solar	5
4	Model Boundary	5
5	Case studies: Application of cost, energy, materials, and time analysis to find an optimal product	7
5.1	Case 1: Long term	7
5.1.1	Analysis: Cost, Energy, and Carbon	8
5.1.2	System Selection	8
5.1.3	Discussion	10
5.2	Case 2: Short term	10
5.2.1	Analysis: Cost, energy	10
5.2.2	System Selection	10
5.2.3	Discussion	10
5.3	Case 3: Instant	11
5.3.1	Analysis: Time, cost, energy	11
5.3.2	System Selection	12
5.3.3	Discussion	12
6	System Comparison: Evacuated tube to flat-pane	13
6.1	Sankey diagrams	13
6.2	Discussion	13
7	Improvements: Diffusion of Innovation	14
8	Conclusion	15
A	Survey	19
B	General impact of different water heater types	20
C	Climate Data	20
D	Modelling the climate	21
E	Modelling solar collectors and tank	21
F	Modelling water transport and pipes	22

G	Modelling boosters	23
H	ACT energy prices	24
I	Modelling hot water usage	24
J	Real systems: Specifications and retail prices	24
K	PERT chart for solar	25
L	Sankey diagrams for 4-minute showers	25
M	MATLAB source code for thermal model	26
N	MATLAB run script for thermal model	42
O	MATLAB source code for hot water survey results processing	50
P	MATLAB code for survey data processing	56

1 Introduction

1.1 Motivation

Heating water is the most significant form of household energy usage, accounting for over 1/3 of total use (Ausgrid, 2013). There are literally hundreds of types and configurations of hot water systems, all with different energy consumption, life-cycle cost, performance, and efficiency. Solar hot water systems can significantly decrease this consumption by drawing energy from the sun. They do not yet make up a large portion of the Canberran market: Out of a survey of 122 ENGN2226 students, only 6 indicated that their place of residence in Canberra used solar hot water (Figure 1). This cannot be entirely dismissed on the basis of the high density of students living on campus, as only 44 live in an apartment/residence. It was also noted that more people are unaware of even the fuel that heats their water than have solar systems (Figure 1).

This ignorance of one of the most energy-intensive household processes is something we must address. This portfolio aims to provide the reader with the necessary information to select the most appropriate system to be used in a Canberran residence, through characterisation of energy requirements, cost, emissions, and performance, as well as increase awareness of wasteful behaviours and system configurations.

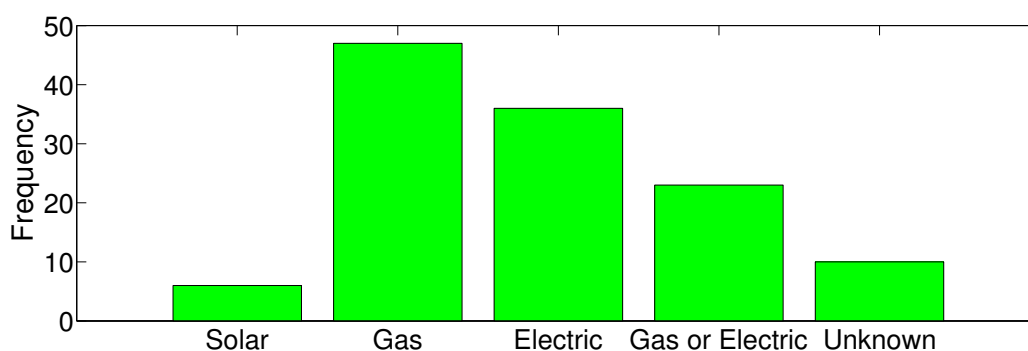


Figure 1: Prevalence of different hot water systems among ENGN2226 Students

1.2 Approach

Because of the breadth of possible customer requirements/habits/scenarios, it is impossible to find a single ideal system for the Canberran customer; instead, we will perform three case-studies. These examples will demonstrate how a customer with certain requirements might select a system based on this analysis.

We use systems analysis techniques to determine the benefits and shortcomings of selected hot water systems and configurations. Recommendations to reduce cost, heat loss, impact, and waiting times are discussed with reference to detailed analysis. Many of these factors can be reduced by considering usage patterns, system selection, and configurations of hot water pipes.

2 System Requirements: Human Factors

Clearly, a hot water system should deliver water to a tap, shower, bath, etc. at an appropriate pressure, temperature, and time. To simplify analysis, we will not consider pressure as a factor, but will treat temperature and time in some depth. The system should also have a low environmental impact and life-cycle cost (LCC).

2.1 Waiting time

Our survey of ENGN2226 students indicated that people are generally satisfied with waiting 10-14 seconds for hot water, but would prefer to wait less than 5 seconds (Figure 2). A comparison of the waiting times between different systems is given in Figure 3. Since it is unlikely that people are acutely aware of how long they might wait for hot water, they were asked to give a value to the nearest 10 seconds, resulting in an error of ± 5 seconds in the data. Figure 3 shows that perceived waiting time is not highly correlated with the type of system. While we might infer that solar has higher waiting times, recall from Figure 1 that this is based on an extremely small sample size. Indeed, we will demonstrate that waiting times have little to do with the type of system you have, but the length of the pipes connecting your tap to the heater. The exceptions to this are continuous systems which circulate hot water constantly through the pipes (Rheem, 2013), and instantaneous electric hot water systems which heat on-demand at the tap (ZIP, 2013). A copy of the survey given to the ENGN2226 class and the code used to process the data may be found in Figure 12 of Appendix A and Appendix O/Appendix P respectively.

Figure 2: Hot water waiting time satisfaction

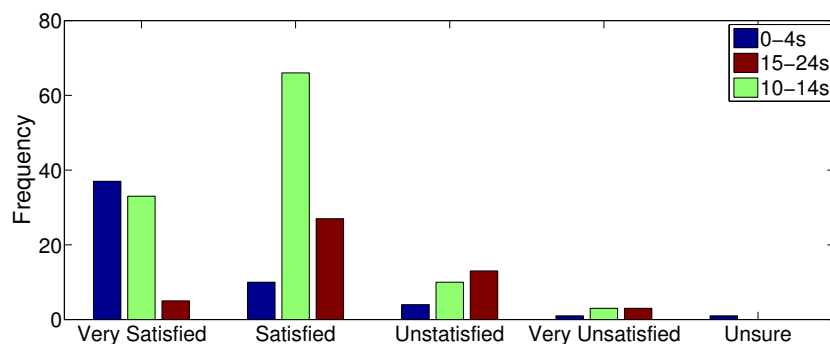
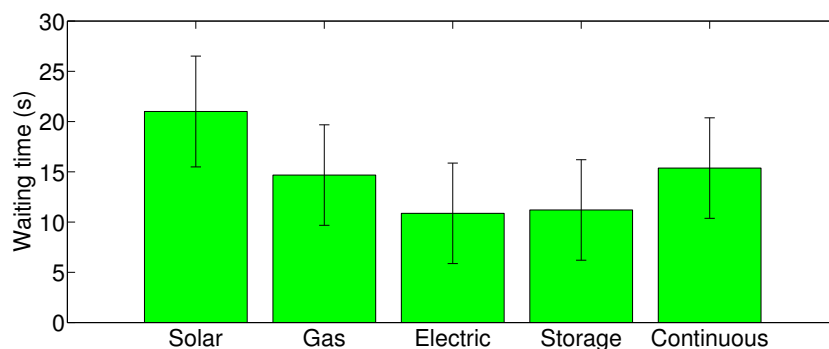


Figure 3: Average waiting times for different systems, with error



2.2 Temperature

Human comfort and human health govern the temperature at which hot water systems operate. Human comfort requires no more than 41°C for showers and washing up; any hotter and we experience pain and/or scalding injuries (TMVA, 2000). However, we cannot set our water heaters to 41°C due to legionella bacteria, which causes potentially fatal legionnaires disease. Legionella grows at temperatures between 20 and 45°C, but dies above 60°C (TMVA, 2000). The Australian standard for hot water systems specifies that water must be heated to certain temperature for a certain amount of time each week (Standards Australia, 2009). These times and temperatures are listed in Appendix G, Table 9, though do not apply if heating from directly from the cold outlet. We comply with these standards in our model as best we can:

If the water in the solar tank does not comply with Table 9, the continuous heating system is set to heat 20°C higher for the entire month (from 50° to 70°). This effectively models a system controlled manually by the user, who would not know the temperature in the tank, but would set the temperature higher during colder months. It does not necessarily apply to storage-boosted systems (i.e. where the heating element is inside the tank), but we do not consider these in our analysis.

We have two different instantaneous systems available in our model: gas and electric. We have assumed that the gas system is a single external unit, but that the electric system is decentralised, and is made up of multiple separate systems, which heat water to only the temperature required by the user.

3 System types: Preliminary time and energy analysis

We have selected several of the most common hot water systems available to perform our analysis upon. Note that we have neglected gas and electric storage heaters, and instead focus on continuous systems and solar storage, as they are higher performance. Indeed, the Australian government is phasing out electric storage heaters as they have a far higher environmental impact than any other type of system (Energy Matters, 2009). Refer to Appendix B for a rough comparison of emissions between several system types.

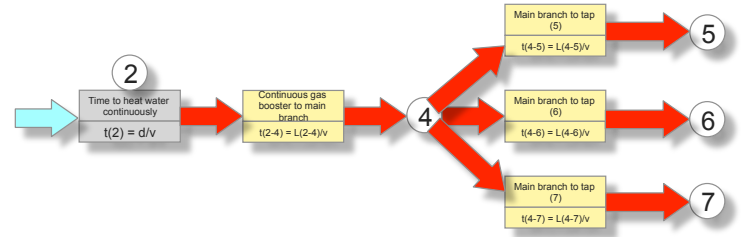
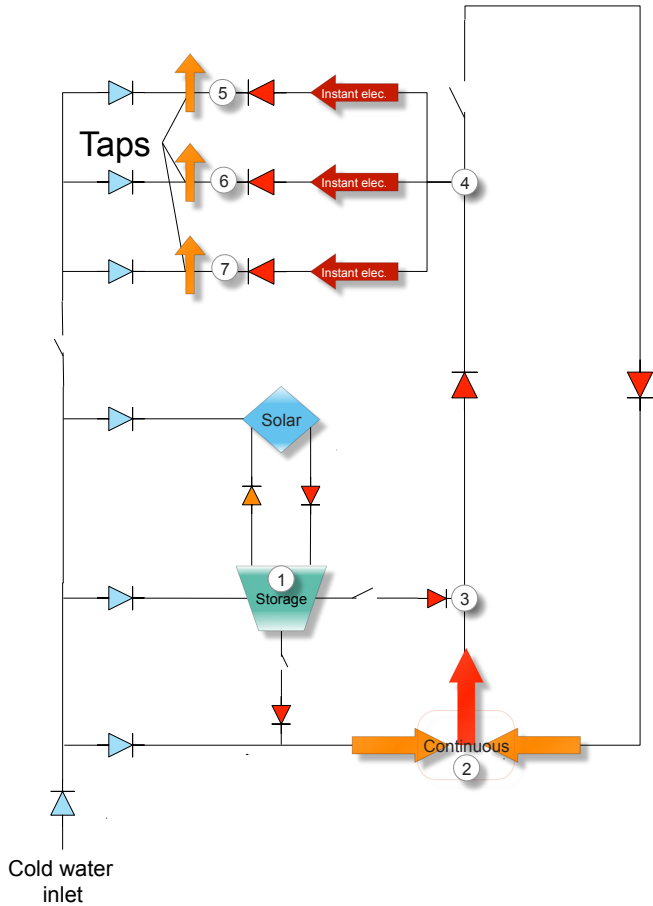
A schematic of how different systems interact with pipelines in a home is given in Figure 4a.

3.1 Continuous gas

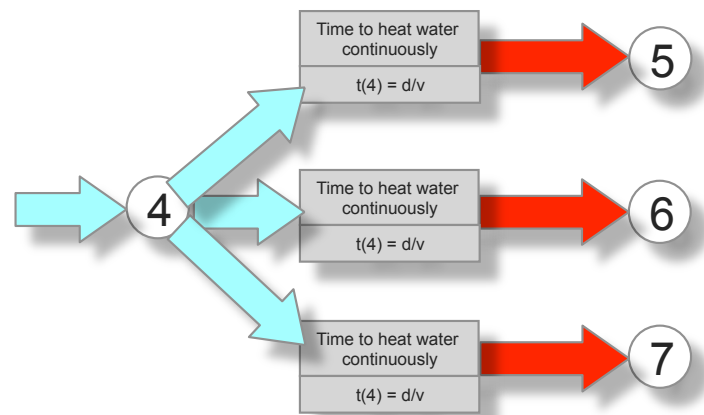
Continuous gas heaters heat water using gas combustion as the water flows through the system in a pipe. A direct quote from Rheem (2013) describes the operation perfectly: "The water heater operates automatically, heating water as it passes through the water heater. When a hot tap is opened, the gas burners ignite to provide immediate heating of the water. The heat produced by the burner is transferred to the water through the heat exchanger. The water is heated to a constant temperature by the automatic adjustment of the gas supply to the burner to suit the water flow rate. The gas burners extinguish when the hot tap is closed." Heat transfer is given by the following well-known relation, which is used profusely in our model:

$$\dot{Q} = \dot{m}c(T_h - T_c) \quad (1)$$

Where \dot{Q} is heat transfer rate in W, \dot{m} is mass flow rate in kg/s, c is specific heat capacity of the water in J/(kg K), T_h is the hot outlet temperature, and T_c is the cold inlet temperature.



(b) PERT chart for continuous gas system to describe waiting time from when tap is turned on until hot water flows out of the tap. $L(a-b)$ denotes the length of the pipe between point a and b with reference to Figure 4a, v is the velocity of water, and d is the length of the pipe inside the heater



(c) PERT chart for instantaneous electric system to describe waiting time from when tap is turned on until hot water flows out of the tap

(a) Water circuit diagram of possible system configurations. Switches designate connections that may be opened or close, depending on what type/configuration the system is. Blue and red arrows indicate cold and hot flow respectively

Figure 4: Water flow diagrams and PERT charts showing how hot water systems interact with pipelines, and what this means for waiting times

The total energy that must be output by the heater is then:

$$\dot{Q}_{actual} = \dot{Q} / \eta_{real}$$

Where η_{real} is the real efficiency of the heater. The real efficiency is the direct ratio between the energy expended in heating the water, and the energy actually transferred to the water: the lower this efficiency, the more heat is wasted as exhaust. For a gas heater under residential use, $\eta \approx 75\%$ (Thomas, 2006).

These systems can either operate on-demand, as described by Rheem (2013), or continuously circulate water, reducing waiting time. With reference to Figure 4a, a PERT chart was created to quantify waiting time (Figure 4b).

In Figure 4b, $d \ll L$ for all L , and we know from Rheem (2013) that the heater does not slow the water down as it passes through. We thus only consider the waiting time due to the water traveling from the system itself to the point of use, which may be anywhere from 5 to 20m in a typical house. Circulating systems eliminate the time associated with point 2-4 in Figure 4b, since they circulate hot water between 2, 4 and back to 2.

3.2 Instant (continuous) electric

We consider instant electric systems to be resistance heaters, localised to each individual tap. This means that multiple separate systems must be purchased for each point/region of use. We apply exactly the same equations for energy analysis as for continuous gas systems, but assume 100% real efficiency. These systems essentially eliminate waiting time as a factor, by heating the water instantaneously at the source (Figure 4c).

3.3 Solar

Solar hot water systems gather heat from the sun by circulating a heat exchanger (HX) fluid through a collector where it heats up, and then back to a storage tank, where heat is exchanged with the cooler water, gradually heating it. See Appendix E for how we quantify this heat transfer rate. There are two types of collectors we treat in this analysis: evacuated tubes and flat panel.

Evacuated tube collectors are sets of long tubes through which HX fluid flows. The tubes are insulated by a vacuum space, encapsulated by glass. Since glass does not transmit long-wave radiation, these tubes are able to absorb short-wave radiation with very low losses from long-wave radiation and conduction. Their shape also allows them to passively track the sun; when the sun hits them at an oblique angle, the spacing between the tubes exposes more active area to the sun. The effect of this is computed via an incidence angle modifier (IAM), given in Table 8 of Appendix E (Ibrahim, Fardoun, Younes and Louahlia-Gualous, 2014).

Flat plate collectors operate in a similar manner, but the collector area is a flat rectangle of heat-absorbing material, encapsulated in a flat sheet of glass. There is no vacuum present, so heat loss is much higher. These systems do not perform as well over a wide range of temperatures or under low solar irradiance (Ibrahim et al., 2014), but are generally cheaper than evacuated tubes (Appendix J, Table 12).

We could also perform PERT analysis for solar; since the water must go from a tank to a booster, and then to taps, we would expect a slightly longer waiting period, though it depends on the specific configuration of the household. See Appendix K, Figure 14.

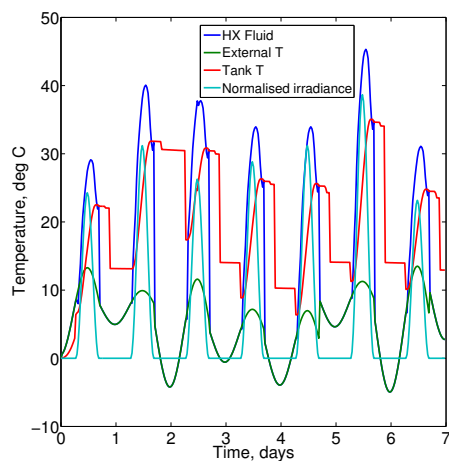
4 Model Boundary

We construct a detailed model in MATLAB to describe key performance metrics of the selected hot water systems. Much of this type of data can be readily found online, but very rigid assumptions are used, so modelling improvements/fluctuations is impossible. Furthermore, some assumptions are often not stated, and yet can have huge effects on the results of analysis. This code is a tool to make very case-specific decisions, and provide the user with a depth of understanding that cannot be found by reading online guides. The code may be found in Appendix M. It is well-commented, but not a complete or robust package; it is useful as a proof-of-concept, but is not ready for real use. We will not explain its inner workings; rather, a model boundary table is provided to give the reader a sense of its scope.

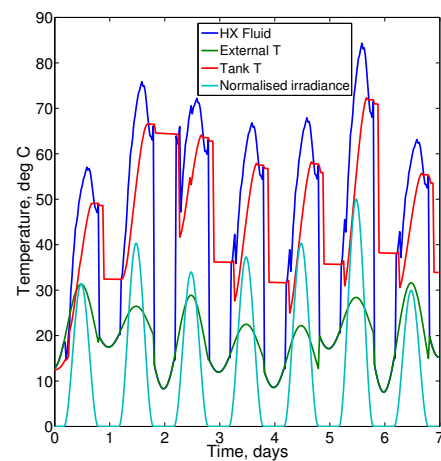
The main strength of the model is its ability to generate the temperature in a solar hot water tank over time. It takes random fluctuations in external conditions into account to accurately model the energy input required of the booster, and/or give the user an impression of un-boosted performance. Figure 5 graphically shows samples of output for summer and winter months.

Table 1: Model boundary table of MATLAB code designed to model energy usage, cost, and impact of various hot water systems for a Canberran household

Endogenous	Exogenous	Excluded
HX fluid flow rate	# people in household	Thermosyphon effects
Emission factors	Booster output temp.	Electric/gas storage systems
Electricity/gas prices (Appendix H)	Number of collectors	Water pressure
Day length	Collector angle/orientation	Split temp. control for gas system
Desired tap temperatures (Appendix I)	Distance of tank to solar collector	Mixing of water in pipes
R-value of tank	Length of pipes from heater to draw points	Stratification in tank (assume well-stirred)
Solar tank temp. (30min time-steps)	Volume of water usage by consumer	Radiation/convection heat transfer (pipes/tank)
Active area of collector	Hot pipe diameter	Aperture areas/IAM factors for different collector manufacturers
Instantaneous efficiency of collectors (Equation (2))	Pipe R-value	Phase change in HX fluid (assume water/glycol mix)
Typical flow rates	Monthly min/max ambient air temps. (Appendix C)	Energy used to drive water pump
Typical usage durations (Appendix I)	Circulating hot water	Electricity/gas price fluctuations/subsidies
Sinusoidal ambient air temp.	Circulating time	Off-grid gas prices
Random variation in temp. and irradiance		Variations in usage outside 1 week
Electricity used by continuous gas		Climates outside Canberra regions
Hot water used in time to get to tap		Standards other than Australian
		Imperfect heat exchange between HX fluid and tank



(a) Winter



(b) Summer

Figure 5: Sample outputs for 1-week periods of the year. Assumptions: 4-person household in Canberra with 44-tube evacuated tube system, 315L storage tank. T denotes temperature, and irradiance is in W/m^2 , normalised to fit on T scale

5 Case studies: Application of cost, energy, materials, and time analysis to find an optimal product

We can use our analytical model of various hot water systems to estimate the energy used over an entire year, based on the climate (see Appendices C and D), and typical usage levels (Appendix I) and patterns. Using energy prices for the Canberra region (Appendix H), we can convert this into an annual running cost.

A report undertaken by the Australian Government (2013) gives us a direct equivalency between energy use and effective greenhouse gas (GHG) emissions for different fuels. For households, these values include emissions due to inefficiency of electric generation plants, transmission losses, fuel transport, mining, etc. It is called the scope 3 emission factor (EF), and it provides us with an effective estimation of the impact of a fuel over its entire life-cycle - from raw processing to local end use: In the ACT/NSW, for electricity, $EF = 293 \text{ kgCO}_2/GJ$, and for main's natural gas, $EF = 12.8 \text{ kgCO}_2/GJ$.

Note that although natural gas may be more inefficient at directly heating water, the impact of heating with electricity is over 20 times higher. Unfortunately, this is not the entire basis upon which customers make purchasing decisions; cost and payback period are also major factors. We perform LCC analysis using retail prices, warranty information and energy analysis of different systems, and attempt to optimise for three sets of customer requirements/scenarios.

We have also performed a materials audit to estimate the embodied energy of a 22-tube evacuated tube system (based on information given by Apricus (2013)). Note that the values given are likely slightly larger than in reality - in particular, we know from Apricus (2013) that the dry weight of the system should be 71.3kg (excluding the mount and tank), but our analysis puts it at 77 kg, even neglecting certain smaller contributors, such as absorber coatings and rubber caps.

Table 2: Embodied energy of evacuated tubes, including mounting frame and 315L tank. All measurements (from which volumes were estimated) from Apricus (2013) for the tubes/mount, and Apricus (2008) for the tank, and all embodied energies/carbons and densities from Hammond and Jones (2008)

Tubes	Embodied energy (MJ/kg)	Embodied carbon (kg CO ₂ /kg)	Volume (m ³)	Density (kg/m ³)	Mass (kg)	Embod. energy (MJ)	Embod. carbon (kg)
Borosilicate glass	21.8	1.27	0.0120	2500	30	652	38
Al (manifold casing)	213.5	11.2		2700	9	1964	103
Copper	70	3.83	0.00311	8600	27	1872	102
Al (mounting)	213.5	11.2	0.00418	2700	11	2408	126
Stainless Steel	56.7	6.15	0.0000346	7850	0.27	15	2
Total (tubes)					77	5143	314
Tank							
Stainless steel (tank)	56.7	6.15	0.00968	7850	38	2163	235
Polyurethane	72.1	3	0.0800	30	2	173	7
Glass	23.5	1.27	0.00831	2500	21	488	26
Total (tank)					61	2824	268

5.1 Case 1: Long term

A family of four is shortly moving into a newly-built house in Canberra, in which they plan to live for over 15 years. They are environmentally conscious, and do not mind paying a bit

extra for a more environmentally-friendly system. What is the best system for them?

5.1.1 Analysis: Cost, Energy, and Carbon

Table 3: Summary of costs, energy usage, and environmental impact of several selected hot water system types and sizings for a 4-person household over a 15-year period. Embodied energy analysis only carried out for evacuated tube system, plus a 315L glass-lined tank. See Table 12 in Appendix J for cost sources. All costs in AUD.

System type	Retail price	Install. cost	STC Rebate	Ann. maint. cost (over 15 years)	Ann. energy cost	Ann. replac. cost (over 15 years)	Embod. energy (MJ)	Embod. carbon (kg)	Ann. non-renew. energy (MJ)	Ann. CO2 Emissions (kg)
x1 Evac (un)	2845	800	0	67	0	0	14273	1100	0	0
x2 Evac (un)	4460	800	0	67	0	0	21814	1508	0	0
x1 Evac (gas)	3845	800	640	67	860	100	14273	1100	28700	388
x2 Evac (gas)	5460	800	1480	67	495	100	21814	1508	16700	225
x1 Flat (gas)	3000	600	480	67	1050	100			35500	473
x2 Flat (gas)	4500	600	1360	67	666	100			21200	302
Continuous gas	1000	300	0	100	810	100			27400	367
Continuous electric	2400	400	0	167	1030	320			20300	5940

Note that the installation prices shown in Table 3 are only estimates, reflecting likely relative costs of each; actual figures depend strongly on individual cases, and it is necessary to obtain an official quote for an accurate value. Solar systems (in particular evacuated tubes) are generally more difficult to install, as they have more parts, etc., and a similar argument is true of a decentralised continuous electric system. Maintenance costs are also estimated in this manner, based on the warranties of the systems, given in Appendix J. Solar tanks contain a sacrificial anode which must be replaced every 5 years (Apricus, 2008), which we assume may cost \$200. Continuous electric systems can suffer from calcium buildup, which may require annual cleaning (Ibrahim et al., 2014). We assume that a plumber might need to be hired every 3 years to clean out these systems (at \$500). Continuous gas systems recommend similar maintenance every 5 years (Rheem, 2013).

The rebate included in Table 3 is based on Small-scale Technology Credits (STCs). These are similar to market shares in that they are created when your system is installed, and may be bought or sold for profit at any given time. They are subject to fluctuating share prices, so we will not model this behaviour. However, one does have the option to sell all one’s STCs upfront for a flat rate of \$40/STC (Clean Energy Regulator, 2014a), which is what the ”rebate” is based on. The number of STCs generated by a system is based on an estimate of the number of MWhrs of electricity saved by the system over a 10-year life, so is higher for larger/more efficient systems. A calculator of generated STC’s can be found online (Clean Energy Regulator, 2014b). Since systems are not generally sold without boosters, we will neglect STC rebates for un-boosted systems, as they will make up the value of the rebate quickly with 0 annual energy costs.

5.1.2 System Selection

The customer wants a cost-effective system over 15 years, which is environmentally friendly. It is clear from Figure 6 that 2 evacuated tube systems (of 22 tubes each, i.e. a 44-tube system) is optimal in terms of cost (Figure 6a) and environmental impact (Figure 6c); cost is \$1400 lower than that of a continuous gas system after 15 years, and total CO_2 emissions are lower by 1344kg (Table 2).

Of course, technically the ”best” system according Figure 6 is a 22-tube un-boosted system. However, this system is not in compliance with the Australian standard for the control

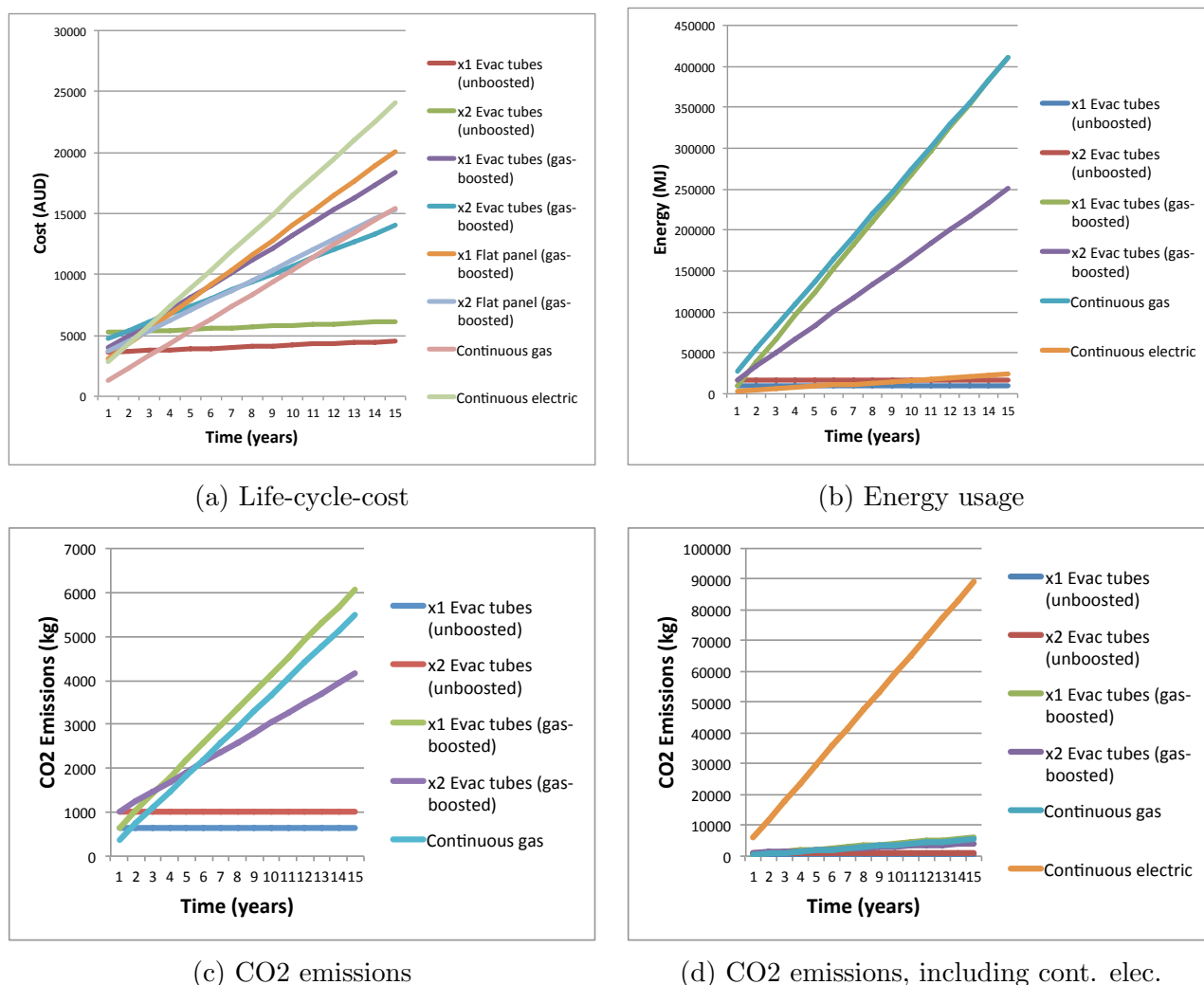


Figure 6: Cost, energy usage, and environmental impact over 15 years for different systems, based on cost analysis from Table 3 and on energy analysis outlined in Table 1

Table 4: Human factors performance of un-boosted solar hot water systems, based on detailed energy analysis

System	Ann. incidences of sub-ideal water temperature	Ann. no. months requiring 70° boost
x1 Evac.	80	9
x2 Evac.	37	4
x1 Flat.	112	12
x2 Flat	51	6

of legionella bacteria (Appendix G). Furthermore, the performance of this (and all other un-boosted systems) in terms of basic human comfort is sub-par (Table 4).

5.1.3 Discussion

A 22 evacuated tube or a single flat-panel system are inappropriate for a family of this size, according to the analysis in Figure 6. Due to the insufficient solar energy from the sun, the hot water tank is merely being warmed to a temperature at which legionella bacteria can proliferate, requiring the gas booster to heat water 20°C higher than would ordinarily be necessary, in accordance with the Australian standard (Appendix G). This causes energy usage to be even higher than if the system were not present.

Figure 6c indicates that the environmental impact (in terms of CO_2 emission) of an evacuated tube hot water system is equivalent of 2-3 years' usage of a continuous gas system. Since we must still boost this system, the actual CO_2 payback period is around 6 years for a 44-tube system, which is significantly shorter than the warranty (Appendix J). Thus in this case, a 44-tube system is carbon-positive, whilst also being cost-effective.

Continuous electric is by far the most expensive and harmful system, yet it uses barely any energy directly; over 15 years, it consumes under 1.5 times the amount of energy as it takes to manufacture a 44 evacuated tube system. The reason for this is the high cost of electricity in Canberra, and the dirtiness of Australia's grid electricity, which derives mostly from coal-burning plants. In the future, we can expect renewables such as photovoltaic (PV) and wind systems to perform a larger (even dominant) role in our grid electricity, driving both cost and emissions down drastically (Blakers, 2014). At this point, solar thermal hot water may become obsolete, as it will be cheaper and likely cleaner to heat with electricity from the sun rather than heat.

5.2 Case 2: Short term

A couple is renting a house in Canberra, in which they plan to live for the next 3-5 years. They currently have a typical continuous gas heater, but want to know if it is worth upgrading to solar, and using their current gas system to boost. Which solar collector is appropriate, if at all? Does this answer change if there are 6 people in this household?

5.2.1 Analysis: Cost, energy

Costs are again taken from Table 3. Since we assume the household already has a gas booster, we take the cost of the booster to be 0, and subtract \$1000 from the cost of each solar system, since the booster is included in the original price.

5.2.2 System Selection

We can clearly see from Figure 7 that solar hot water needs far longer than 5 years to represent an intelligent investment, regardless of the magnitude of usage. We will not show energy and impact analysis, as this was treated in the previous section in some detail, and it is far less relevant here; at a loss of at least \$1100, there is clearly no motivation to move to solar for a period less than 5 years. By far the most cost-effective is the household's current system, continuous gas, regardless of the number of people in the household.

5.2.3 Discussion

Since the family is renting, it is unlikely that they would receive any extra money when they moved out; they cannot sell the system, and the landlord has limited motivation to absorb the cost: the landlord would then simply have to charge more for rent, as they would not

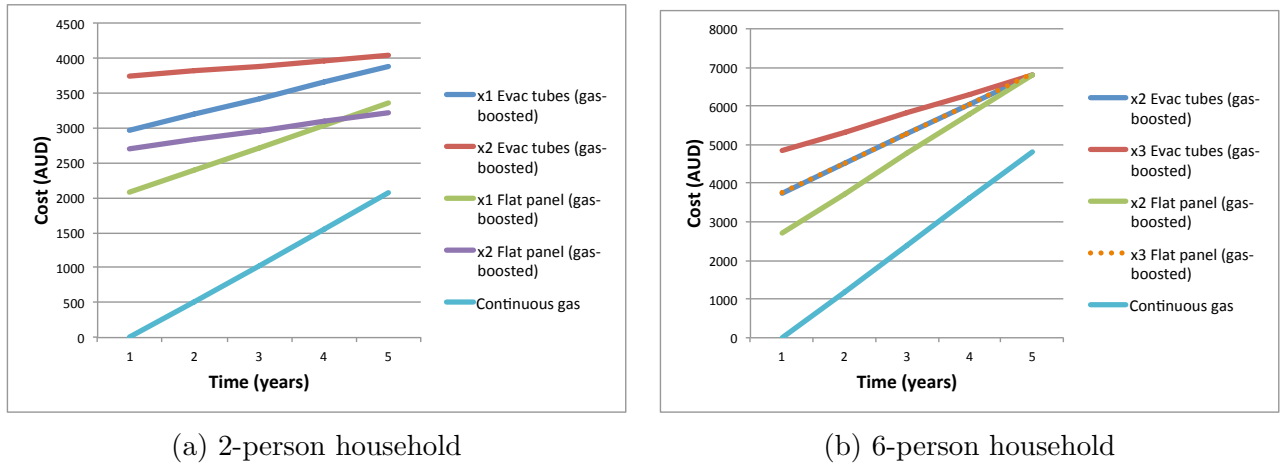


Figure 7: Cost over 5 years for case 2

necessarily wish to pay utility fees for their tenants.

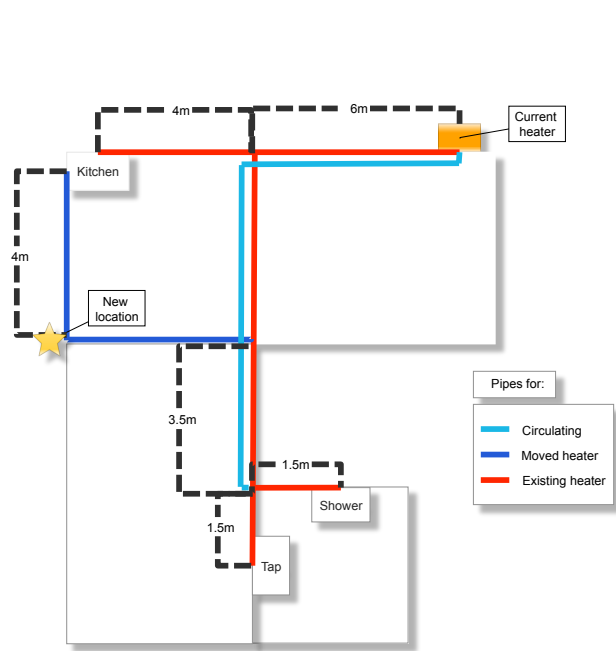
An interesting note is that for a 2-person household, 2 flat-panel systems is likely the best long-term solution, with the lowest cost slope, and payback after 8 years (apart from 44 evacuate tubes, but this remains more expensive than the flat-panel for a total of 18 years at the current rate, which is longer than system warranty), whereas 3 evacuated tubes is best for a 6-person household, though the payback is in approximately another 3 years. This indicates that flat-panel systems may be more appropriate for small-scale usage, since they are cheaper initially and can perform almost as well under lower loads.

5.3 Case 3: Instant

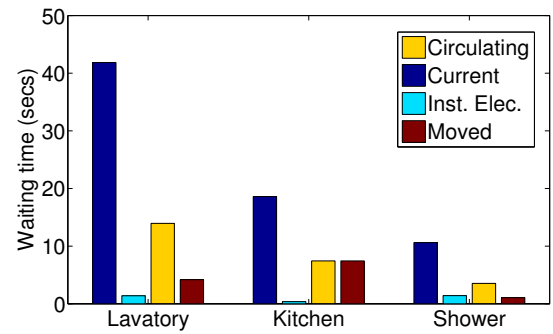
A couple is fed up with waiting up to 20 seconds for hot water out of their kitchen tap, 10 for their showers, and over 40 for the bathroom sink. They are currently using a continuous gas system, located as shown in Figure 8a, and are considering 3 options, and know approximate upfront prices of each:

1. Purchase 2 instant electric hot water systems for bathroom and kitchen: \$1800 (Table 13)
2. Upgrade their existing system to circulate hot water around the main line for 4 hours per day (in the morning and evening when usage is highest): \$500
3. Bring in a plumber to move their system to the point indicated in Figure 8a: \$900

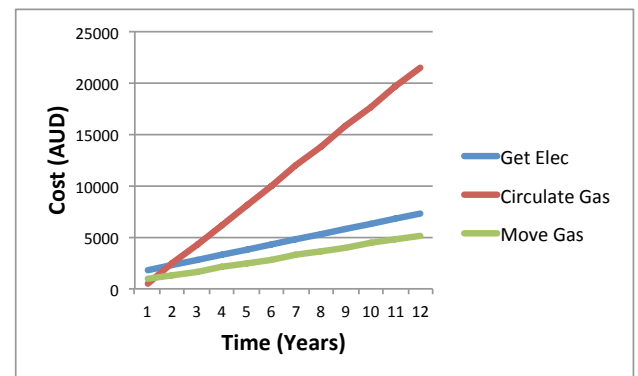
5.3.1 Analysis: Time, cost, energy



(a) Pipe arrangement for the 3 options considered in case 3, including dimensions in metres



(b) Waiting times at key usage points for each scenario



(c) Solution to time analysis problem by cost end energy analysis

Figure 8: Case 3 time analysis problem and solution

5.3.2 System Selection

We can see from Section 5.3.1 that moving the gas system is cheapest, and offers the 2nd best solution to the waiting time problem. Purchasing an electric system is only marginally more expensive (around \$500 on average), and offers much shorter waiting times. We know that an electric system may be less reliable, since it only has a 12 month warranty (ZIP, 2013), although it would be new. Figure 2 indicates that most people would be satisfied with the waiting times associated with simply moving the system, but would be very satisfied by those provided by a new instantaneous electric system. There is no clear winner here, though if money is no particular object, and environmental impact is not a factor, the electric system is the better choice.

5.3.3 Discussion

This cannot be said for circulating gas; cost is 3.3 times higher after 5 years compared to simply moving the system, and performance is worse, since the circulating pipes cannot go all the way to each tap (if they did, this would increase installation cost, and the length of the circuit, which would drive ongoing cost up further).

The reason for this high cost is the loss of heat in the 13.5m long circulating pipe, which the gas heater must replenish for 4 hours each day.

If the family were looking for a more environmentally friendly solution, recall that grid

electricity emits over 20 times more CO₂ per unit energy used than gas, which makes it by far the most harmful (at least on our current grid). Figure 6d shows the severity of this effect for case 1, but the same principle applies here.

6 System Comparison: Evacuated tube to flat-pane

6.1 Sankey diagrams

It is instructive to understand the physical reasons for the differences between evacuated tube and flat-panel in energy usage, cost, impact, etc. that were presented in Section 5. Our energy model will allow us to identify areas of the design that could easily be improved, as will be discussed in Section 7. We will continue to treat analysis for a 4-person household in greatest detail. For the scenario in case 1, we generate Sankey diagrams to visually compare the energy flows of a gas-boosted 44 evacuated tube system with those of a gas-boosted double flat-panel system.

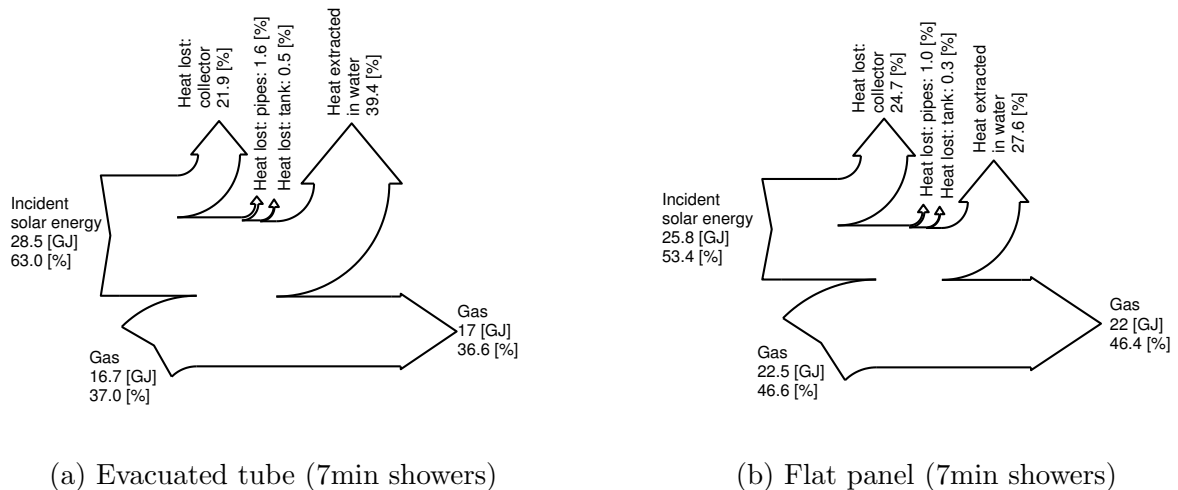


Figure 9: Sankey diagrams demonstrating performance differences between evacuated tube and flat panel solar collectors. Diagrams produced in MATLAB using code developed by Spelling (2010)

6.2 Discussion

We see from Section 6.1 that the evacuated tube collector loses 12.8% (absolute, i.e. $\frac{24.7-21.9}{21.9} \times 100$) less solar energy than the flat panel system. This can be attributed to the excellent conduction control offered by the vacuum between the ambient air and the absorber. This translates to a relatively larger reduction in the extra energy delivered by the gas booster (27% absolute). This is attributed to the non-linear effect of keeping the water in the tank hot enough to control legionella.

Direct optical and thermal losses in the collector are significant, amounting to nearly 35% of incident radiation for evacuated tubes, and over 46% for flat panel. Evacuated tubes are the better design, but to increase optical/thermal efficiency further would require more complex processing and structures. Since evacuated tubes are just barely cost-effective over

extended time periods, any specific improvements we might suggest would not necessarily benefit the system as a whole. Losses in the pipes and tank are low, due to high R-values of both items. These R-values are easy to obtain, using foam insulation, we cannot improve this to any significant benefit.

The temperature in the tank is highly dependent on the volume of water drawn from it, since we assume hot water is immediately replaced with cold. This may not be true for all system types, but it is certainly true for thermosyphons, which are passive and rely on pressure/temperature differences to control flow. They are closed and must keep constant pressure, so when hot water is drawn from the tank, it is replaced by cold simply to equalise pressure in the tank. With such a system, lowering hot water usage is vital, and will have an extreme effect on its performance (refer to Section 7).

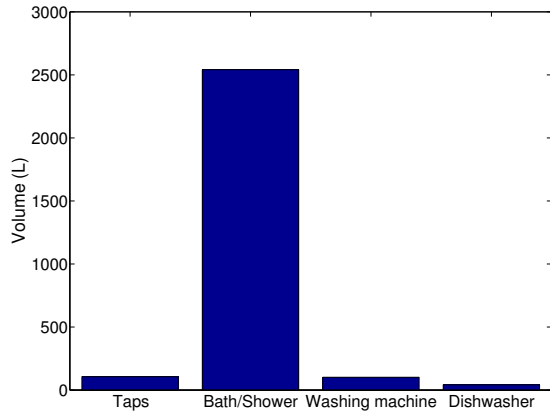
7 Improvements: Diffusion of Innovation

Although solar hot water has been around for many years, uptake is not high in Canberra (Figure 1). We know from Section 5 that solar energy is low impact, but its high initial cost, and the necessity of boosting is preventing it from becoming cost-effective (or even carbon-positive) on a reasonable timeframe for most households: for a typical 4-person household, cost payback is around 11 years, and carbon payback is 6 years. We use the Pareto principle to identify some simple changes to solar hot water systems to shorten these payback periods. We have identified water usage and embodied carbon to be key contributors to long energy paybacks for solar hot water systems. Section 7 shows the contributions of various factors to water usage and embodied carbon. Figure 10a assumes 7 minute showers with a standard shower head outputting 15 L/min, which according to WELS (2014c) is the Australian standard. We also assume every person in the household showers once per day. With these reasonable assumptions, showering/bathing accounts for around 25 times the rest of household usage put together.

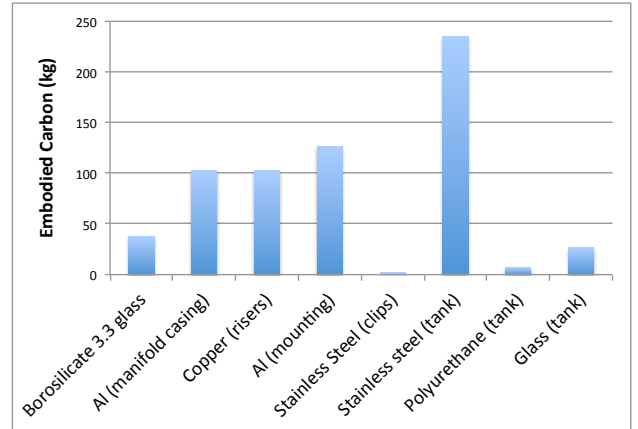
The two most carbon-intensive elements of evacuated tube systems are the stainless steel tank and the aluminium mounting frame. It is interesting to note that neither of these items are intrinsic to the collector itself, which actually has very low embodied carbon, being made of relatively innocuous materials such as glass, copper, and relatively small amounts of aluminium. Ideally, we might naively suggest some alternate materials with low embodied energy to replace the stainless steel/aluminium components; this would not be meaningful, as there are obviously good reasons for using these materials, such as a cost/performance trade-off. We have already shown in Section 5.1.3 that the embodied carbon of evacuated tube systems is equivalent to 2-3 years' usage of a continuous gas system. This is a small timeframe compared to the system's warranty of at least 10 years (Appendix J). Substituting other materials may decrease embodied energy, but be detrimental to the failure rate, and thus the warranty. We limit ourselves then to a modest alteration: according to Aprius (2013), the aluminium mounting frame can be made of steel instead, which has lower embodied energy/carbon than aluminium. Since this is recommended by the manufacturer, we can safely assume that it will not significantly effect other factors.

The following alterations have been made to our original assumptions:

- Daily shower lengths reduced from 7 mins to 4 mins
- Mounting frame changed from aluminium to stainless steel



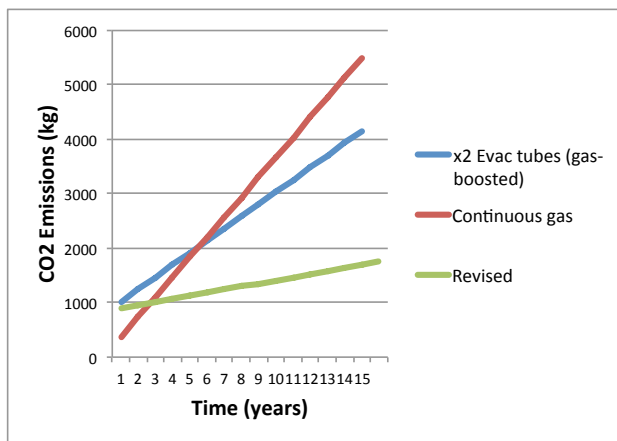
(a) Typical weekly levels of hot water usage for a 4-person household (7 min showers)



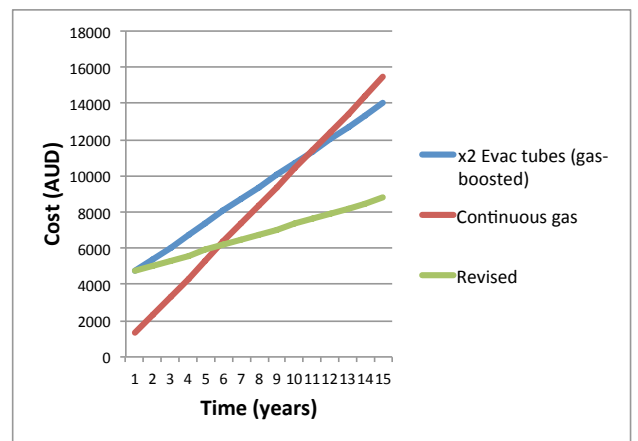
(b) Breakdown of embodied carbon in evacuated tube systems

Figure 10: Bar plots of household water usage and embodied carbon

With these simple improvements to the system and changes to usage patterns, we have shortened cost payback from 11 years to 6, and carbon payback from 6 years to 3.



(a) Revised carbon payback



(b) Revised cost payback

Figure 11: Gas-boosted evacuated tube cost and carbon payback curves: Comparing the original curves from Figure 6 with those generated based on improvements suggested in Section 7

8 Conclusion

The benefits of solar hot water systems can be quite 'fluid', and it is often difficult for households to assess whether they are worthwhile installing. In general, solar hot water systems offer significant cost and environmental impact reduction over a period shorter than their warranty, provided that the system is sized appropriately to the level of usage. It is hoped that this portfolio informs the reader such that they are able to make intelligent purchasing and usage decisions based on a systems approach to analysis.

References

ACTEWAGL (2014a). Our ACT electricity prices.

URL: <http://www.actewagl.com.au/Product-and-services/Offers-and-prices/Prices/Residential/ACT/Electricity-prices.aspx>

ACTEWAGL (2014b). Our ACT natural gas prices.

URL: <http://www.actewagl.com.au/Product-and-services/Offers-and-prices/Prices/Residential/ACT/Natural-gas-prices.aspx>

Apricus (2008). Apricus Australia Glass Lined Steel Solar Water Storage Tank, *Technical report*.

URL: <http://www.energymatters.com.au/images/apricus/apricus-gl-tank-manual.pdf>

Apricus (2013). Product Overview: AP Evacuated Tube Solar Collector, *Technical report*.

URL: http://www.apricus.com/upload/userfiles/downloads/Apricus_AP_Solar_Collector_Overview

Ausgrid (2013). Ausgrid, *Technical report*.

URL: [http://www.ausgrid.com.au/media/Files/Ways to save/Energy efficiency brochures/Ausgrid_Hot water_201213.pdf](http://www.ausgrid.com.au/media/Files/Ways%20to%20save/Energy%20efficiency%20brochures/Ausgrid_Hot%20water_201213.pdf)

Australian Government (2013). Australian National Greenhouse Accounts, *Technical Report July*, Department of Industry, Innovation, Climate Change, Science, Research and Tertiary Education.

URL: http://www.climatechange.gov.au/sites/climatechange/files/documents/07_2013/national-greenhouse-accounts-factors-july-2013.pdf

Blakers, A. (2014). The renewable energy revolution.

Bosch, R. (2012). Domestic Gas Hot Water Systems, *Technical Report October*.

URL: [http://www.bosch-climate.com.au/files/201303270013410.Bosch_dom_2013\(march\)v2.pdf](http://www.bosch-climate.com.au/files/201303270013410.Bosch_dom_2013(march)v2.pdf)

Bureau of Meteorology (n.d.). Climate Information for Canberra Aero.

Clean Energy Regulator (2014a). Creating STCs for solar water heater and heat pump systems.

URL: <http://ret.cleanenergyregulator.gov.au/Certificates/Small-scale-Technology-Certificates/Creating-STCs—hot-water-systems/creating-swh>

Clean Energy Regulator (2014b). Solar water heater STC registry.

URL: <https://www.rec-registry.gov.au/rec-registry/app/calculators/swh-stc-calculator>

Elgas (2014). Hot Water System Reviews & Comparison by Type & Price.

URL: <http://www.elgas.com.au/blog/507-hot-water-heater-systems-compare-review-type-kind-best>

Energy Matters (2009). Australia To Commence Electric Hot Water System Phase Out In 2010.

URL: <http://www.energymatters.com.au/renewable-news/em706/>

Hammond, P. G. and Jones, C. (2008). INVENTORY OF CARBON & ENERGY (ICE).

URL: [http://www.uea.ac.uk/e680/energy/NBS-M016/ICE Version 1.6a.pdf](http://www.uea.ac.uk/e680/energy/NBS-M016/ICE%20Version%201.6a.pdf)

Hot Water Professionals (n.d.). Electric Continuous.

URL: <http://www.hotwaterprofessionals.com.au/shop/residential/7/Electric-Continuous>

Ibrahim, O., Fardoun, F., Younes, R. and Louahlia-Gualous, H. (2014). Review of water-heating systems: General selection approach based on energy and environmental aspects, *Building and Environment* **72**: 259–286.

URL: <http://linkinghub.elsevier.com/retrieve/pii/S0360132313002709>

Pye, J. (2013). *ENGN2222 Thermal Energy Systems course notes*.

Rheem (2013). Owners Guide and Installation InstructionsL Continuous Flow Gas Water Heater, *Technical report*.

Spelling, J. (2010). drawSankey.

URL: <http://www.mathworks.com.au/matlabcentral/fileexchange/26573-drawsankey>

Standards Australia (2003). Plumbing and drainage, Part 4: Heated water services.

Standards Australia (2009). Australian Standard Authorization requirements for plumbing products Water heaters and hot-water storage tanks.

URL: http://r.b5z.net/i/u/10115377/f/AustralianStandards/AS_3498-2009.pdf

SunMaxx Solar (2010). TitanPowerPlus SU-2 Flat Plate Collector, *Technical report*.

URL: <http://www.sunmaxxsolar.com/data-titanpowerplus-su24>

Sustainability Victoria (2009). Large Scale Solar Thermal Systems Design Handbook, *Technical Report December*.

URL: <http://www.sustainability.vic.gov.au//media/resources/documents/publicationsandresearch/knowledgearchive/smallscalerenewabletechnology/commercialandindustrialsolarwaterheaters/largescalesolarthermaldesignhandbookdec2009.pdf>

THERMANN (2013). Evacuated tube solar hot water systems: Thermann technical manual, *Technical Report August*.

URL: <http://www.reece.com.au/new/pdf/products/Evacuated-Tube-Electric-Technical-Manual.pdf>

Thomas, M. (2006). Reducing GHG emissions through efficient water heating technologies, *EIC Climate Change Technology*.

URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4057306

TMVA (2000). Recommended Code of Practice for Safe Water Temperatures, *Technical report*, Thermostatic Mixing Valve Manufacturers Association.

WELS (2014a). Clothes Washing Machine Criteria.

URL: <https://apps5a.ris.environment.gov.au/wels-public/search-product-load.do?search=true>

WELS (2014b). Dishwashers Criteria.

URL: <https://apps5a.ris.environment.gov.au/wels-public/search-product-load.do?search=true>

WELS (2014c). Water Efficiency.

URL: *<http://www.waterrating.gov.au/consumers/water-efficiency>*

ZIP (2013). Zip Instantaneous Hot Water C Series, *Technical report*.

URL: *http://www.zipindustries.com/SiteMedia/W3SVC1283/Uploads/Documents/Zip_IHW_C_Series_Spec_Sheet.pdf*

Appendix A Survey

Household hot water survey

This survey aims to evaluate which hot water systems are common amongst Canberra residents, and how well they perform qualitatively. For multiple choice questions, please check ONE box only. This survey should take less than 5 minutes to complete.

1. What type of hot water system is used in your primary place of residence in the Canberra region? If unsure, consult Figure 1.

- Solar
- Gas
- Electric
- Not sure: either gas or electric
- Don't know
- Other:.....

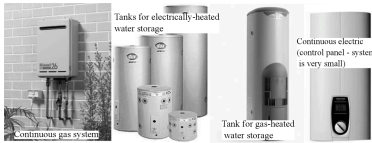
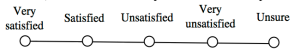


Figure 1: Examples of non-solar hot water systems

2. If you answered ELECTRIC OR GAS (or NOT SURE) to question 1, please specify the type of system:

- Continuous (no tank storage: heating elements switch on when hot water tap is turned on)
- Storage (water stored in one or two insulated tanks. Heated gradually)
- Don't know
- Other:.....

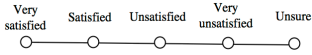
3. Overall, how satisfied are you with the time you have to wait for hot shower water?



How many seconds (to the nearest 10 seconds) would you estimate you wait?

.....

4. How satisfied are you with how long you wait for hot tap water?



How many seconds (to the nearest 10 seconds) would you estimate you wait?

.....

5. Does your household ever lose hot water overnight during winter months? Please choose the answer that best describes the frequency of this event.

- Never
- More than three times a week
- Once a week
- Once a month

6. Over the entire year, how often does your system lose hot water?

- Never
- Once a week
- Once a month
- Once every three months
- Once a year

Questions 7-9 are concerned with SOLAR hot water systems. If you did NOT answer SOLAR to QUESTION 1, skip to QUESTION 10.

7. If you answered SOLAR to question 1, then please specify the type of solar hot water system (refer to Figure 2 if unsure):

- Evacuated tube
- Flat panel
- Parabolic reflector
- Don't know
- Other:.....

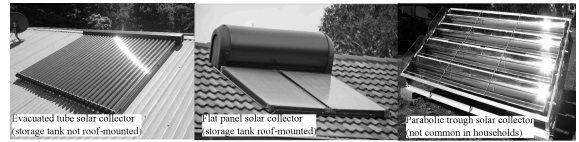


Figure 2: Examples of different types of solar collectors

8. If you answered SOLAR to question 1, does your system have a booster to keep water hot during long periods of cold or cloud cover?

- Yes, gas
- Yes, electric
- No
- Don't know
- Other:.....

9. If you answered SOLAR to question 1, please specify where your storage tank is located

- Roof-mounted
- Indoors
- Outside, but on the ground
- Don't know

10. Do you live in student residence/an apartment, or in a house?

- Apartment/residence
- House

IF YOU ANSWERED "APARTMENT/RESIDENCE" TO QUESTION 10, DO NOT ANSWER THE REMAINING QUESTIONS.

11. How many people (including yourself) live in your household? Please write a number on the dotted line

.....

12. Approximately how often is the dishwasher used in your household?

- Never/don't own one
- Twice a day
- Once a day
- Three times a week
- Twice a week
- Less than once a week

Figure 12: Survey questions given to ENGN2226 class

B General impact of different water heater types

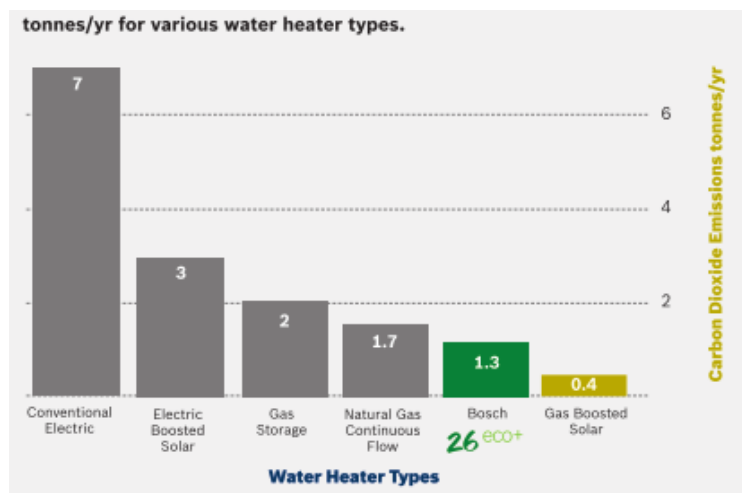


Figure 13: Estimated CO2 emissions based on average Victorian usage (Bosch, 2012)

C Climate Data

Table 5: Average daily solar flux, Φ , in a plane by month for Canberra, and day length to the nearest hour. Flux in MJ/m^2 per day for a horizontal plane, and a north-facing vertical plane (Pye, 2013)

Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Horizontal	28.6	23.7	19.2	13.8	9.8	7.9	9.0	11.9	16.5	21.5	24.7	26.9
Vertical	10.5	12.7	15.3	15.5	13.9	12.3	13.6	15	15.2	13.6	11.0	9.9
Day length	15	14	13	12	10	10	10	11	12	13	14	14

Table 6: Average maximum and minimum daily temperatures in Canberra by month, as well as the range of possible fluctuation from these averages. All in $^{\circ}C$ (Bureau of Meteorology, n.d.)

Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Average max	28	27.5	24	20	16	12	11	12.5	17	19.5	22	27
Average min	12.5	12.5	10.5	7	3	1	0	1	3	6.5	9	10.5
Average max range	13	14.5	14	12	10	8	9	10.5	12	12.5	17	12
Average min range	10	10	10	10	10	10	10	10	10	10	10	10

D Modelling the climate

If the average daily solar flux is given by Φ , then the average daily azimuth angle for each month, θ , is :

$$\theta = \arctan \left(\frac{\Phi_{horiz}}{\Phi_{vert}} \right)$$

Where Φ is given per month in Table 5.

The solar irradiation intensity, G during the day is calculated in Watts for a discrete timestep, modulated by a cosine function, and a random modulation between 0.5 and 1.5, R :

$$G = I_{avg} \left(1 - \cos \left(\frac{2\pi t}{T_{day}} \right) \right) R$$

Where T_{day} is the length of a day in seconds, and G_{avg} is the average irradiance in W, and:

$$G_{avg} = \frac{\Phi_{vert}}{\cos \theta} \sin(\theta + \alpha)$$

Where α is the mount angle of the collector, assuming it is north-facing.

The external temperature also follows a sinusoidal curve, with the maximum and minimum temperatures modulated each day by a random number, based on the average max and min range in Table 6:

$$T_{ext} = \frac{T_{max} + T_{min}}{2} - \frac{T_{max} - T_{min}}{2} \cos \left(\frac{2\pi t}{3600 \times 24} \right)$$

E Modelling solar collectors and tank

The value of \dot{Q} from Equation (1) for a solar collector is given by the simple expression:

$$\dot{Q} = A_{aperture} G \eta$$

Where $A_{aperture}$ is the aperture area of the collector (standard), G is the incident global irradiance (in W/m^2), and η is defined as the ratio of incident solar irradiance power to power delivered to HX fluid in the collector. For glazed flat plate or evacuated tube solar collectors, it is given as a function of experimental data provided by manufacturers, and local climactic conditions:

$$\eta = \left(\eta_o - a_1 \frac{t_m - t_a}{G} - a_2 \frac{(t_m - t_a)^2}{G} \right) IAM \quad (2)$$

Where η_o is the optical efficiency provided by the manufacturer, G is again the incident solar irradiation on the slope of the collector, a_1 and a_2 are coefficients provided by the manufacturer, t_a is the outside ambient temperature, t_m is the average fluid temperature in the collector (Sustainability Victoria, 2009), and IAM is the transversal incidence angle modifier - a measure of the collector's ability to track the sun (Apricus, 2013). IAM is given in Table 8

Table 7: Efficiency metrics provided by manufacturers for flat panel (SunMaxx Solar, 2010) and evacuated tube (Apricus, 2013) collectors

Metric	η_0	a_1	a_2
Flat	0.801	3.64	0.0112
Evacuated tube	0.687	1.505	0.0111

Table 8: Transversal IAM for flat panel and evacuated tube collectors (Apricus, 2013)

Angle (°)	0	10	20	30	40	50	60	70	80	90
Flat	1.00	0.98	0.94	0.87	0.77	0.64	0.50	0.34	0.17	0.00
Evac tube	1.00	1.02	1.08	1.18	1.35	1.47	1.39	1.57	0.95	0.00

Heat loss in a pipe section or tank is given by:

$$Q = \frac{\Delta T}{R} \quad (3)$$

Where R is the R-value (thermal resistance value). For a water tank, it is given in units of m^2K/W , and for a pipe section it is given mK/W .

F Modelling water transport and pipes

Typical volumetric flow rates for various draw types in L/min (Thomas, 2006):

- Lavatory faucet: 3.8
- Bath Tub: 15
- Shower: 15 (Australian standard flow (WELS, 2014c))
- Washing Machine: 7.5
- Dishwasher: 5.7
- Kitchen Sink: 5.7

We calculate the speed of water flowing in the pipes in m/s simply by:

$$v = \frac{\dot{V}}{\pi d^2/4}$$

Where \dot{V} is the volumetric flow rate in m^3/s , and d is the diameter of the pipe in m. The mass flow rate is given by:

$$\dot{m} = \rho \dot{V}$$

Where ρ is the density of water; $\rho = 1000 \text{ kg/m}^3$.

We can use eq. (3) on a small section of pipe to determine the temperature gradient throughout the pipe, and get a more accurate value for heat loss. We neglect heat transfer between adjacent sections, and assume transfer is only between a given section and the air.

Canberra is in insulation region C of Australia, which means that external pipes for solar heaters must have a minimum R-value of 1 mK/W, and internal pipes require minimum R-value of 0.3 mK/W (Standards Australia, 2003).

G Modelling boosters

We use eq. (1) to estimate the amount of power required to heat water from some given initial temperature to another given final temperature. If the continuous (a.k.a. instantaneous) system is operating alone, the initial temperature is given as the ambient air temperature, which we have already shown how to calculate. If the system is boosting a solar heater, the input temperature is the solar tank temperature at the time at which water is drawn.

All systems must comply with the Australian standard for prevention of legionnaires disease. The following is copied directly from Standards Australia (2009):

Means shall be provided to inhibit the growth of Legionella bacteria in potable water within the water-heating appliance. A water-heating appliance will be deemed to comply with this requirement if it satisfies at least one of the following requirement:

- (i) For a storage water heater, heat pump water heater or solar water heater, at least 90% of the stored water is heated to 60C for at least one single period of not less than 32 min in each 7 day period.
- (ii) For a heat pump water heater or solar water heater, at least 45% of the water contained within the storage tank is heated to a minimum of 60C daily.
- (iii) The water heater is operated such that all water is subjected to a temperature dependent minimum exposure period as specified in Table 1 within 7 days before passing through the water-heating appliances heated water outlet

Table 9: Minimum exposure period

Temperature	Minimum exposure period
70°C or greater	1 s
66 °C	2 min
60°C	32 min
55°C	6 h

- (iv) The water heater is an instantaneous water heater or a water heater with a heat exchanger or a combination of both, containing less than a total of 10 L of potable water that is intended to be supplied directly from the cold water supply or from a storage or solar water heater that complies with Items (i), (ii) or (iii) above.
- (v) The water heater is a storage solar pre-heater that is intended to supply another water heater that complies with Items (i), (ii) or (iii) above.

Table 10: Instantaneous booster output temperatures

Average monthly solar tank temperature	Instantaneous booster output temperature
$T < 55^{\circ}C$	$70^{\circ}C$
$55 < T < 60^{\circ}C$	$60^{\circ}C$
$T > 60^{\circ}C$	$50^{\circ}C$
Cold (i.e. no solar)	$50^{\circ}C$ (or as required for inst. elec)

Most gas systems also use some electrical power while they are running for startup/control, etc. They also have an electric resistance heater which prevents water from freezing inside them during the winter when they are not in use. The power associated with these features vary between systems, but for an average gas system by Rheem (2013):

- Anti-frost off: 62 W
- Anti-frost on: 144 W

H ACT energy prices

- Cost of electricity the ACT: AUD\$0.18304 per kWh (ACTEWAGL, 2014a)
- Cost of natural gas in the ACT: AUD\$0.029535 per MJ (ACTEWAGL, 2014b)

I Modelling hot water usage

Table 11: Standard total water use, and total use time of various forms of household hot water use. Estimated based on experiment unless otherwise cited. Total use time is defined at the time water is actually running. It may be based on estimation of actual time (as in the case of a shower), or on a known flow rate, and total use (as in the case of a dishwasher)

Draw point	Bathroom tap	Kitchen tap	Shower	Bath	Washing machine (WELS, 2014a)	Dishwasher (WELS, 2014b)
Water use (L)	0.32	5	105	100	50	13
Use time	5s	53s	7min	6.7min	6.7min	2.3min
Ideal temperature ($^{\circ}C$) (TMVA, 2000)	41	41	41	44	40	50
Assumption	Wash hands	1 wash-up for 4 people	Standard	Standard	5kg clothes	Full load

J Real systems: Specifications and retail prices

*Gas booster costs around \$1000 (Elgas, 2014), and is included in the above prices.

*Instant electric price assumes 2 separate systems must be purchased: one High-flow for shower/bath (around \$1200), and one low-flow for kitchen (around \$600) (Hot Water Professionals, n.d.).

Table 12: Specifications and prices for real solar collectors*. All data from Elgas (2014) unless otherwise stated

System type	Number of modules	Total aperture area	Approximate retail price (AUD)	Minimum Warranty
Evacuated tube (gas boosted)	1	2.07 (THERMANN, 2013)	3845	10 years (Apricus, 2013)
Evacuated tube (gas boosted)	2	4.14 (THERMANN, 2013)	5460	10 years (Apricus, 2013)
Flat panel (gas boosted)	1	1.874(SunMaxx Solar, 2010)	3000	12 years(SunMaxx Solar, 2010)
Flat panel (gas boosted)	2	3.748(SunMaxx Solar, 2010)	4000	12 years4(SunMaxx Solar, 2010)

Table 13: Summary of pricing and warranty information for non-solar hot water systems considered in analysis

System type	Approximate retail price (AUD)	Warranty
Continuous gas	1000 (Elgas, 2014)	5 years (Rheem, 2013)
Instant electric	1800*	1 year (ZIP, 2013)

K PERT chart for solar

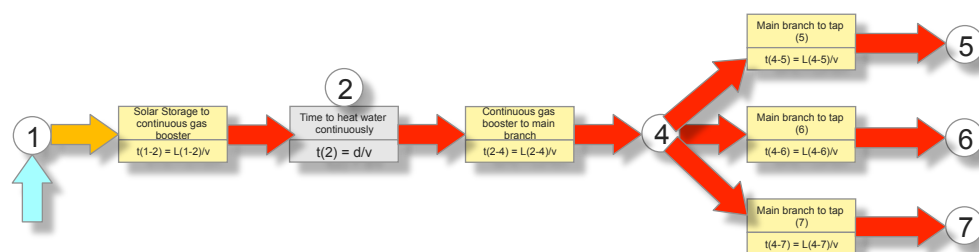
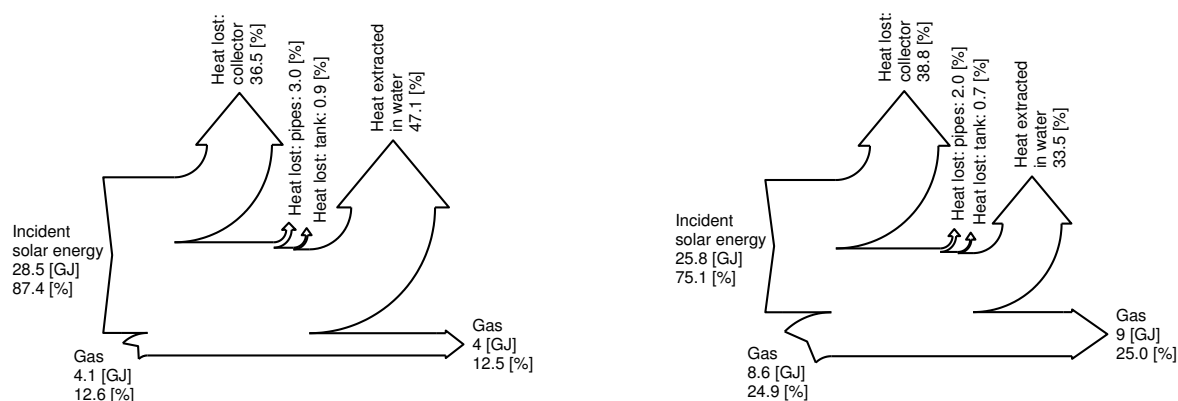


Figure 14: PERT chart for solar system to describe waiting time from when tap is turned on until hot water flows out of the tap

L Sankey diagrams for 4-minute showers



(a) Evacuated tube (4min showers)

(b) Flat panel (4min showers)

Figure 15: Sankey diagrams demonstrating the effect of taking 4 minute showers instead of 7 on the performance of evacuated tube system

M MATLAB source code for thermal model

```
classdef HWSYS
```

```
    methods(Static = true)
```

```
        function loss=loss_pipe(T_int,T_ext,Rval_sect,len_sect)
```

```
            %Calculates the heat loss in W through a section of pipe of  
            %length
```

```
            %len_sect, and R-value Rval_sect.
```

```
            R_th=Rval_sect/len_sect;
```

```
            loss=(T_int-T_ext)/R_th; %W
```

```
        end
```

```
        function mass=mass_pipe(len_sect,diam,rho_w)
```

```
            %Calculates the mass of water contained in a pipe of length  
            %len_sect
```

```
            volume=(len_sect*pi*diam^2)/4;
```

```
            mass=volume*rho_w;
```

```
        end
```

```
        function [T_int_sect,Q_loss]=avg_internal_temp(c_w,rho_w,...
```

```
            T_heater,T_ext,R_val,diam,speed,LENGTH)
```

```
            %Estimates the internal temperature through the pipe sections
```

```
            %Break pipe into section of length equal to speed in m/s,
```

```
            %i.e. the distance traveled by the hot water in 1s. There  
            %will be LENGTH/speed sections.
```

```
            %Also outputs the total loss along the length of the pipe, in W
```

```
            %We need a looping vector:
```

```
            pipe_sects=1:LENGTH/speed;
```

```
            %And the length of each section:
```

```

len_sect=speed;
%Each section will have an R-value equal to the total
%R-value divided by the number of sections (assume
%negligible heat transfer between adjacent sections):
Rval_sect=R_val/length(pipe_sects);
%Each section has mass:
mass_sect=HWSYS.mass_pipe(len_sect,diam,rho_w);
%Generate an empty vector for T_in_sect for reassignment:
T_int_sect=zeros(1,length(pipe_sects));
%The first section will have a temperature of T_heater:
T_int_sect(1)=T_heater;
Q_loss_sect=zeros(1,length(pipe_sects)-1);
%Loop through all other sections
for sect=pipe_sects(2:length(pipe_sects))
    %Calculate loss from one section to the next, which is
    %equivalent to 1 second to the other
    Q_loss_sect(sect-1)=HWSYS.loss_pipe(T_int_sect(sect-1),...
        T_ext,Rval_sect,len_sect);
    T_int_sect(sect)=T_int_sect(sect-1)-Q_loss_sect(sect-1)/...
        (mass_sect*c_w);
end
Q_loss=sum(Q_loss_sect);
end

```

```

function travel_time=time_pipe_accel(LENGTH,acceleration)
    travel_time=sqrt(2*LENGTH/acceleration); %s
end

```

```

function acceleration=avg_accel_pipe(pressure,diam,rho_w,LENGTH...
    ,friction)
    %Calculates the average acceleration of water in a pipe of a
    %given LENGTH, assuming constant pressure and frictional force
    %along its LENGTH, and that frictional force is independent of
    %the velocity of the flow

    %pressure in Pa
    %diam and LENGTH in m
    %rho_w in kg/m^3
    %friction in N/m^2
    acceleration=(diam*pressure-4*friction*LENGTH)/(diam*rho_w*...
        LENGTH); %m/s^2
end

```

```

function speed=speed_pipe_accel(acceleration,LENGTH)
    speed=sqrt(2*LENGTH*acceleration);
end

```

```

function travel_time=time_pipe_flow_rate(speed,LENGTH)
    travel_time=LENGTH/speed;
end

function speed=speed_pipe_flow_rate(flow_rate,diam)
    %flow rate in m^3/s
    %diam in m
    area=(pi*diam^2)/4;
    speed=flow_rate/area; %m/s
end

function R_val=R_pipe(inner_diam,outer_diam,k)
    %Calculates the thermal resistance of an insulated pipe per
    %unit LENGTH. Neglecting convective effects
    %Calculate conductive resistance/LENGTH
    R_val=(log(outer_diam/inner_diam)/(2*pi*k)); %(K m)/W
end

function power=power_inst_or_cont(T_i,T_o,rate,rho_w,c_w,real_eff)
    %Calculates the power required to heat water from temperature
    %T-i to T_o at a given flow rate and real use efficiency
    %real_eff as a ratio between 0 and 1
    %c_w = SPECIFIC heat capacity of water
    %rate is volumetric flow in m^3/s
    %All inputs in SI units
    power=rho_w*rate*c_w*(T_o-T_i)/real_eff; %W
end

function loss=storage_loss(T_int,T_ext,R_tank,A_tank)
    loss=(T_int-T_ext)/(R_tank*A_tank);
end

function cost=energy_running_cost(energy_source,energy_used,time)
    %energy_source a 1 or 2-element vector. First element is energy
    %in Joules, second element only applies to gas - it is the
    %wattage of the gas heater to run a defroster, pump, etc.
    %Usually 62W, but can be as high as 146W
    cost_per_kWh_elec=0.18304; %AUD (ACTEWAGL)
    cost_per_J_elec=cost_per_kWh_elec/(3600*1000);
    cost_per_MJ_gas=0.029535; %AUD, (ACTEWAGL)
    cost_per_J_gas=cost_per_MJ_gas/10^6;
    if strcmp(energy_source(1),'G')== 1
        elec_energy_used=energy_source(2)*time;
        cost=cost_per_J_gas*energy_used+elec_energy_used*...
            cost_per_J_elec;
    elseif strcmp(energy_source(1),'E')== 1
        cost=cost_per_J_elec*energy_used;
    end
end

```

```

end

function cost=water_running_cost(rate,time)
    %Rate in m^3/s, time in s
    cost_per_kL=2.64; %AUD (ACTEW)
    %1 kL is the same as 1 m^3, so:
    cost=cost_per_kL*rate*time;
end

function CO2_emissions=impact_running(energy_source,...
    energy_used,time)
    %energy_source a 1 or 2-element vector. First element is energy
    %in Joules, second element only applies to gas - it is the
    %wattage of the gas heater to run a defroster, pump, etc.
    %Usually 62W, but can be as high as 144W
    %energy_used in J, CO2_emissions in kg
    kg_CO2_per_J_gas=13*10^-9; %(Australian gov. report)
    kg_CO2_per_J_elec=293*10^-9; %(Australian gov. report)
    if strcmp(energy_source(1),'G') == 1
        elec_energy_used=energy_source(2)*time;
        CO2_emissions=energy_used*kg_CO2_per_J_gas+...
            elec_energy_used*kg_CO2_per_J_elec;
    elseif strcmp(energy_source(1),'E') == 1
        CO2_emissions=energy_used*kg_CO2_per_J_elec;
    end
end

function del_T_tank=chang_temp_tank(C,Q_deliv,Q_lost,time)
    %Calculates temperature differential of water tank over some
    %time, based on the heat delivered, lost, and the heat
    %capacity. All in SI units
    del_T_tank=((Q_deliv-Q_lost)*time)/C;
end

function Q_lost=tank_draw_loss(flow_rate,c_w,T_tank,T_cold)
    %Calculates the heat lost from a water storage tank if water is
    %drawn out at a given rate, and replaced with cold water.
    %Flow rate in kg/s
    Q_lost=flow_rate*c_w*(T_tank-T_cold);
end

function eff=collector_efficiency(T_ext,T_fluid,a1,a2,eta0,irrad,...
    collector,day_time,day_length)
    %Takes climactic data and standard experimental data provided
    %by manufacturers and outputs an efficiency value. This
    %efficiency is the ratio of incident irradiance to power
    %delivered to the HX fluid

```



```

    if irrad == 0
        eff=0;
    else
        del_T=T_fluid - T_ext;
        IAM=HWSYS.IAM_factor(day_time,collector,day_length);
        eff=(eta0-a1*(del_T/irrad)-a2*((del_T^2)/irrad))*IAM;
    end
end

function Q_sol=solar_gain(irradiance,apert_A,eff)
    %Irradiance in W/m^2. aper_A is the aperture area of s solar
    %collector, or the roof space area. eff, efficiency, is the
    %ratio between incident power and energy delivered to the HX
    %fluid. It includes conductive, radiative, and convective
    %losses.
    Q_sol=irradiance*apert_A*eff;
end

function del_T_fluid=T_fluid_out(flow_rate,c,Q_in)
    %Calculates the rate of change in temperature of fluid
    %traveling with some flow rate, exposed to some constant
    %heat input.
    %flow_rate in kg/s
    del_T_fluid=Q_in/(flow_rate*c);
end

function irradiance=avg_irradiance_collector(month,mount_angle)
    %Calculates average daily irradiance in W per m^2 incident on a
    %solar collector mounted at mount_angle. Based on data for a
    %vertical plane and the azimuth angle for given Canberran month.
    %month must be an integer between 1 and 12, mount angle in
    %degrees
    flux_horiz=[28.6,23.7,19.2,13.8,9.8,7.9,9.0,11.9,16.5,...
        21.5,24.7,26.9]; % in MJ/m^2/day (Pye, 2013)
    flux_vert=[10.5,12.7,15.3,15.5,13.9,12.3,13.6,15,15.2,13.6,...
        11.0,9.9];
    azimuth=atand(flux_horiz(month)/flux_vert(month));
    day_lengths=[15,14,13,12,10,10,10,11,12,13,14,14];...
        %Day length in hours (Pye, 2013)
    day_length=day_lengths(month);
    flux_vert_month=flux_vert(month);
    flux_max=flux_vert_month/cosd(azimuth);
    flux_mount=flux_max*sind(mount_angle+azimuth);
    %Convert MJ/m^2/day to W/m^2
    irradiance=(flux_mount*10^6)/(day_length*3600);
end

function T_ext=ext_temp(T_max,T_min,time)

```

```

    %Time in seconds, all temps in deg C. Begins in the middle of
    %the night at T_min.
    T_diff=(T_max-T_min)/2;
    T_avg=(T_max+T_min)/2;
    f=1/86400;
    T_ext=T_avg-T_diff*cos(2*pi*f*time);
end

function T_ext_switch=switch_temp(T_max,T_min,night_length)
    %Takes input of night_length in HOURS, and outputs the external
    %temperature at which we switch from day to night or vice
    %versa, i.e. when we switch solar gain on or off.
    F=night_length/24;
    f=1/86400;
    T_switch_arg=(2+F)/(2*f);
    T_ext_switch=HWSYS.ext_temp(T_max,T_min,T_switch_arg);
end

function irradiance_t=irradiance_funct_t(avg_irradiance,...
    day_length,time)
    %Outputs incident irradiance as a function of some time, based
    %on the day length and the average irradiance given by
    %avg_irradiance_collector.
    %day_length must be given in hours, time in seconds, and
    %avg_irradiance in W/m^2 to output irradiance_t in W/m^2:
    T_d=day_length*3600; %Convert day_length to seconds
    irradiance_t=avg_irradiance*(1-cos((2*pi*time)/T_d));
end

function [day_length,night_length]=daynightlengths(month)
    day_lengths=[15,14,13,12,10,10,10,11,12,13,14,14];...
    %Day length in hours (Pye, 2013)
    day_length=day_lengths(month);
    night_length=24-day_length;
end

function time=waiting_time(draw_rate,diam,LENGTH)
    speed=HWSYS.speed_pipe_flow_rate(draw_rate,diam);
    time=LENGTH/speed;
end

function time_index=time_24hr_index(timestep,time_24hr)
    %This function takes a time in 24hr form and converts it to an
    %index of a vector, assuming that this vector begins at
    %midnight and each subsequent entry is larger by a time
    %"timestep" (which is in seconds)
    seconds_since_midnight=time_24hr*3600;
    timesteps_since_midnight=seconds_since_midnight/timestep;

```

```

        time_index=round(timesteps_since_midnight);
end

function [weekly_draw_rate_L,total_use_tot,total_time_draw,...
        actual_rate,usage]=weekly_draw(time_168hr_usage,timestep,...
        diam,length_pipes)
%Takes a cell of cells input, where each cell contains the time,
%usage type, and multiplier for each hot water draw throughout a
%given day. This function outputs draw rate for each timestep
%in L/sec. If the timestep is larger than the ordinary time
%interval over which draw takes place, this function normalises
%the draw rate to compensate.
%use_time is the time hot water actually flows for, whereas
%total_time is the use time plus the waiting time, which is
%primarily a function of the flow rate, and length and diameter
%of the pipes
weekly_draw_rate_L=zeros(1,round((24*7*3600)/timestep));
total_usage=zeros(1,length(time_168hr_usage));
usage=zeros(1,length(length_pipes)-2);
for draw=1:length(time_168hr_usage)
    type=time_168hr_usage{draw}{2};
    [draw_rate,use_time,wait_time]=HWSYS.draw_info(type,...
        length_pipes,diam);
    total_time=wait_time+use_time;
    draw_rate=draw_rate*time_168hr_usage{draw}{4}; %rate multip
    total_time=total_time*time_168hr_usage{draw}{3};%time multip
    total_usage(draw)=draw_rate*total_time;
    %Get the total usage for each type:
    if strcmp(type,'L')==1 || strcmp(type,'K')==1
        %lump kitchen and bathroom sink together
        usage(1)=usage(1)+total_usage(draw);
    elseif strcmp(type,'S')==1 || strcmp(type,'B')==1
        %lump bath & shower together
        usage(2)=usage(2)+total_usage(draw);
    elseif strcmp(type,'W')==1
        usage(3)=usage(3)+total_usage(draw);
    elseif strcmp(type,'D')==1
        usage(4)=usage(4)+total_usage(draw);
    end

    number_steps=ceil(total_time/timestep);
    normalised_rate=total_usage(draw)/(timestep*number_steps);
    %Find timestep index of time given
    %Index of starting time:
    index_start=HWSYS.time_24hr_index(timestep,...
        time_168hr_usage{draw}{1});
    total_time_hrs=total_time/3600;

```

```

        index_end=HWSYS.time_24hr_index(timestep,time_168hr_usage...
            {draw}-{1}+total_time_hrs);
    %Enter normalised rates over relevant number of timesteps
    %Takes overlapping draws into account by adding previous rate
    %to new rate for any given timestep. We use a normalised rate
    %for each timestep so draw rate is constant but total use is
    %unchanged
    weekly_draw_rate_L(index_start:index_end)=...
        weekly_draw_rate_L(index_start:index_end)...
        +linspace(normalised_rate,normalised_rate,...
            length(index_start:index_end));
    %Also want to output the time taken for each draw
    total_time_draw(draw)=total_time;
    %Also output the actual rate because we need it for
    %continuous heaters
    actual_rate(draw)=draw_rate;
end
total_use_tot=sum(total_usage);
end

function [max_avg,min_avg]=avg_temp_range_canberra(month)
    avg_max_Ts=[28,27.5,24,20,16,12,11,12.5,17,19.5,22,27];
    avg_min_Ts=[12.5,12.5,10.5,7,3,1,0,1,3,6.5,9,10.5];
    max_avg=avg_max_Ts(month);
    min_avg=avg_min_Ts(month);
end

function [T_max,T_min]=max_min_temps_rand(month,days)
    %Temperature ranges by month (BoM)
    ranges_max=[13,14.5,14,12,10,8,9,10.5,12,12.5,17,12];
    ranges_min=linspace(10,10,12);
    %size of range of possible max daily temperatures
    range_max=ranges_max(month);
    %size of range of possible min daily temperatures
    range_min=ranges_min(month);
    %Average daily maxs and mins
    [max_avg,min_avg]=HWSYS.avg_temp_range_canberra(month);
    T_max_min=max_avg-0.5*range_max; %Lowest possible max temperature

    T_min_min=min_avg-0.5*range_min; %Lowest possible min temperature
    random_temps_max=...
        load('/Users/bennettschneider/Documents/ANU/Semester 2, 2014/Systems/
    random_temps_min=...
        load('/Users/bennettschneider/Documents/ANU/Semester 2, 2014/Systems/
    T_max=T_max_min+range_max*random_temps_max.random_set_temps_max;
    T_min=T_min_min+range_min*random_temps_min.random_set_temps_min;
end

```

```

function switch_index=switch_times(T_ext_avg,switch_temp)
    %Outputs the index of switching times based on a switching
    %temperature and average temperature fluctuations. Allows
    %randomness in actual temperatures without changing day lengths
    switch_index=zeros(1,length(T_ext_avg));
    for t=2:length(T_ext_avg)
        if min(T_ext_avg(t),T_ext_avg(t-1)) <= switch_temp &&...
            max(T_ext_avg(t),T_ext_avg(t-1)) >= switch_temp
            switch_index(t)=1;
        end
    end
end

function [T_tank_draw,Q_deliv_tot,Q_pipe12_tot,Q_pipe34_tot,...
    Q_lost_draw_tot,Q_lost_cond_tot,T_fluid3,T_ext,T_tank,...
    irradiance]=solar(c_HX,rho_HX,R_ext,R_tank,c_w,C_tank,...
    apert_A,eta0,a1,a2,diam_in,length34,length12,days,...
    runtime,timestep,timesteps,...
    month,mount_angle,tank_location,draw_rate,collector)

%Flow rate of HX fluid between tank and collector:
flow_rate_min=1.5; %L/min
flow_rate=(flow_rate_min*rho_HX)/(60*1000); %kg/s
flow_rate_m3=flow_rate_min/(60*rho_HX);
speed=HWSYS.speed_pipe_flow_rate(flow_rate_m3,diam_in);

[day_length,night_length]=HWSYS.daynightlengths(month);%In hours

[T_max_avg,T_min_avg]=HWSYS.avg_temp_range_canberra(month);

T_ext_avg=HWSYS.ext_temp(T_max_avg,T_min_avg,0:timestep:runtime);

T_ext_switch=HWSYS.switch_temp(T_max_avg,T_min_avg,night_length);

switch_index=HWSYS.switch_times(T_ext_avg,T_ext_switch);
%Gives a random max and min temperature about a certain
%range each day
[T_max,T_min]=HWSYS.max_min_temps_rand(month,days+1);

T_inside=20;
random_irrad=...
    load('/Users/bennettschneider/Documents/ANU/Semester 2, 2014/Systems/
shade_factor=0.5+random_irrad.random_set_irrad; %rand(1,days+1);
T_ext=zeros(1,round(timesteps));
T_ext(1)=T_min_avg(1);

Q_deliv=zeros(1,round(timesteps));

```

```

Q_lost_tank=zeros(1,round(timesteps));
Q_lost_draw=zeros(1,round(timesteps));
Q_lost_cond=zeros(1,round(timesteps));
efficiency=zeros(1,round(timesteps));
T_fluid2=zeros(1,round(timesteps));
T_fluid3=zeros(1,round(timesteps));
T_fluid23=zeros(1,round(timesteps));
T_fluid4=zeros(1,round(timesteps));
T_tank=zeros(1,round(timesteps));
irradiance=zeros(1,round(timesteps));
del_T_tank=zeros(1,round(timesteps));

T_fluid2(1)=T_ext(1);
T_fluid3(1)=T_ext(1);
T_fluid23(1)=T_ext(1);
T_fluid4(1)=T_ext(1);
T_tank(1)=T_ext(1);
Q_pipe34=zeros(1,round(timesteps));
Q_pipe12=zeros(1,round(timesteps));
avg_irradiance=HWSYS.avg_irradiance_collector(month,mount_angle);

%We begin in the middle of the night, so:
efficiency(1)=HWSYS.collector_efficiency(T_ext(1),T_ext(1),...
    a1,a2,eta0,0,collector,0,day_length);
day='n';
day_no=1; %Count number of days
draw_count=0; %Count number of times water is drawn
%This variable counts the amount of time that has passed
%since daybreak in seconds
day_time=0;
for t=2:round(timesteps);

    if switch_index(t) == 1
        %Change from day to night or night to day
        if irradiance(t-1) == 0 %If it was night before
            %Change to day
            day='d';
            day_time=0;
        else
            %Change to night
            day='n';
            day_no=day_no+1; %Add a day
        end
    end

    T_ext(t)=HWSYS.ext_temp(T_max(day_no),T_min(day_no),...
        timestep*t); %External temperature

```

```

T_cold=T_ext(t);

if strcmp(tank_location,'Inside')== 1
    T_ext_tank=T_inside;
elseif strcmp(tank_location,'Outside')== 1
    T_ext_tank=T_ext(t);
else
    fprintf('\nError in tank_location string\n')
end

if strcmp(day,'d')== 1
    %If it is daytime, we have irradiance
    %timestep has passed since the last loop
    day_time=day_time+timestep;
    irradiance(t)=shade_factor(day_no)*...
        HWSYS.irradiance_funct_t(avg_irradiance,...
            day_length,day_time);
elseif strcmp(day,'n')== 1
    irradiance(t)=0;
end

efficiency(t)=HWSYS.collector_efficiency(T_ext(t),...
    T_fluid23(t-1),a1,a2,eta0,irradiance(t),collector,...
    day_time,day_length);
Q_sol=HWSYS.solar_gain(irradiance(t),apert_A,efficiency(t)); %W
%Change in temperature of fluid due to the solar collector
del_T_fluid23=HWSYS.T_fluid_out(flow_rate,c_HX,Q_sol);%degC
%T_fluid is the average temperature of the water from the solar
%collector, so in t=timestep, the temperature of the outlet HX fluid
%will be:
T_fluid3(t)=T_fluid2(t-1)+del_T_fluid23;
[T_int_sect34,Q_pipe34(t)]=HWSYS.avg_internal_temp...
    (c_HX,rho_HX,T_fluid3(t),T_ext(t),R_ext,diam_in,speed,length34);
%Final value in T_int_sect34 is the temperature at point 4
T_fluid4(t)=T_int_sect34(length(T_int_sect34));
Q_deliv(t)=flow_rate*c_HX*(T_fluid4(t)-T_tank(t-1));
if T_fluid3(t) < T_tank(t-1)
    %Stop the flow of HX fluid, so repress
    %pipe loss and heat delivery
    Q_pipe34(t)=0;
    Q_deliv(t)=0;
end

Q_lost_draw(t)=HWSYS.tank_draw_loss(draw_rate(t),c_w,...
    T_tank(t-1),T_cold);
Q_lost_cond(t)=(T_tank(t-1)-T_ext_tank)/R_tank;
Q_lost_tank(t)=Q_lost_draw(t)+Q_lost_cond(t);
%Find the initial temperature of the tank when water is drawn

```

```

    if draw_rate(t) > 0
        draw_count=draw_count+1;
        T_tank_draw(draw_count)=T_tank(t-1);
    end
    %Change in temperature of the tank over some timestep
    del_T_tank(t)=HWSYS.chang_temp_tank(C_tank,Q_deliv(t),...
        Q_lost_tank(t),timestep);
    T_tank(t)=T_tank(t-1)+del_T_tank(t);

    [T_int_sect12,Q_pipe12(t)]=HWSYS.avg_internal_temp(c_HX,...
        rho_HX,T_tank(t),T_ext(t),R_ext,diam_in,speed,length12);
    %Find the new input temperature to the solar collector
    T_fluid2(t)=T_int_sect12(length(T_int_sect12));
    %Average temperature if HX fluid in the collector
    %is average of in and out
    T_fluid23(t)=(T_fluid3(t)+T_fluid2(t))/2;
    %Stop the flow of HX fluid, so rerpess pipe loss and assume
    %water in all pipes quickly goes to T_ext
    if irradiance(t) == 0
        Q_pipe12(t)=0;
        T_fluid2(t)=T_ext(t);
        T_fluid23(t)=T_ext(t);
        T_fluid3(t)=T_ext(t);
        T_fluid4(t)=T_ext(t);
    end
end
Q_deliv_tot=sum(Q_deliv*timestep);
Q_pipe12_tot=sum(Q_pipe12*timestep);
Q_pipe34_tot=sum(Q_pipe34*timestep);
Q_lost_draw_tot=sum(Q_lost_draw*timestep);
Q_lost_cond_tot=sum(Q_lost_cond*timestep);
end

function tap_temps=tap_temp(time_168hr_usage)
    %This function takes the type of usage for each separate draw
    %rate given by time_24hr_usage, and outputs an ideal set
    %temperature for the process:
    tap_temps=zeros(1,length(time_168hr_usage));
    for draw=1:length(time_168hr_usage)
        type=time_168hr_usage{draw}{2};
        if strcmp(type,'L')== 1
            tap_temps(draw)=41;
        elseif strcmp(type,'K')== 1
            tap_temps(draw)=41;
        elseif strcmp(type,'S')== 1
            tap_temps(draw)=41;
        elseif strcmp(type,'B')== 1
            tap_temps(draw)=44;
        end
    end
end

```



```

        elseif strcmp(type,'W')== 1
            tap_temps(draw)=40;
        elseif strcmp(type,'D')== 1
            tap_temps(draw)=50;
        end
    end
end

function time_xhr_usage=draw_times_general(time_24hr_usage,days)
    %Takes time_24_hr_usage and extends it to work over a number of
    %days, adjusting the hours given for subsequent entries
    %Duplicate over number of days, minus 1 for existing day
    extended_times= repmat(time_24hr_usage,1,days-1);
    for draws=length(time_24hr_usage)+1:length(extended_times)
        %Day associated with draw
        day=floor(draws/length(time_24hr_usage)-0.0001)+1;
        draws_index=draws-length(time_24hr_usage)*(day-1);
        %Replace the current hours with the same hours plus 24*days
        %that have passed
        extended_times{draws}{1}=...
            time_24hr_usage{draws_index}{1}+24*day;
    end
    time_xhr_usage=extended_times;
end

function IAM=IAM_factor(day_time,collector,day_length)
    %day_time is in seconds
    %We estimate the transversal IAM for the evacuated tube system
    %using the following data:
    if strcmp(collector,'evac')== 1
        IAMs=[0,0.95,1.57,1.39,1.47,1.35,1.18,1.08,1.02,1,...
            1.02,1.08,1.18,1.35,1.47,1.39,1.57,0.95,0];
    elseif strcmp(collector,'flat')== 1
        IAMs=cosd(-90:10:90);
    end
    %This is non-linear, so we will just assign a value based
    %on which is closer in the day
    day_fraction=day_time/(3600*day_length); %day_length in hours
    %Find the index corresponding to the closest fraction of
    %the day, e.g. if day_fraction = 0.33, day_fraction_rounded
    %will output 3, because it is in the the first 3/10 of the
    %day, and so we index the 3rd entry of IAM
    day_fraction_rounded=round(day_fraction*length(IAMs));
    IAM=IAMs(day_fraction_rounded);
end

function [energy_used_gas_tot,energy_used_elec_tot,...
    energy_cost_gas_tot,energy_cost_elec_tot,...

```

```

    CO2_emissions_gas_tot,CO2_emissions_elec_tot,...
    warm_count,legionella_count]...
    =inst_gas_elec_boost(actual_rate,draw_rate,...
    draw_rate_time,T_ext,T_source,real_eff_gas,...
    real_eff_elec,T_heater,tap_temps,rho_w,c_w,...
    circulating,length_circuit,diam_in,R_int,weeks,...
    T_inside,circ_time)
%Outputs the total energy, cost, and greenhouse emissions due
%to continuous gas and electric systems over a month. Can also
%estimate the average monthly increase in these factors for a
%circulating system using continuous gas.

if strcmp(circulating,'yes')== 1
    for test=1:2;
        %Calculate extra heat loss due to circulating hot water
        flow_circ=4; %L/min
        flow_circ_m3=flow_circ/(60*1000); %m^3/s
        speed=HWSYS.speed_pipe_flow_rate(flow_circ_m3,diam_in);
        [T_int_circ,Q_loss]=HWSYS.avg_internal_temp(c_w,...
            rho_w,T_heater,T_inside,R_int,diam_in,speed,...
            length_circuit);
        if max(T_int_circ)-min(T_int_circ) > 5
            T_heater=60;
        end
    end
    %Power required to keep this temperature profile is equal
    %to the heat power lost from the system
    month_time=circ_time*3600*7*weeks;
    %Monthly energy input for circulating system
    Q_circ=Q_loss*month_time;
else
    Q_circ=0;
end

energy_used_gas=zeros(1,length(draw_rate_time));
energy_used_elec=zeros(1,length(draw_rate_time));
energy_cost_gas=zeros(1,length(draw_rate_time));
energy_cost_elec=zeros(1,length(draw_rate_time));
CO2_emissions_gas=zeros(1,length(draw_rate_time));
CO2_emissions_elec=zeros(1,length(draw_rate_time));
draw_count=0;
legionella_count=0;
warm_count=0;

for t=1:length(T_ext);

    if draw_rate(t) > 0
        draw_count=draw_count+1;
    end
end

```

```

%Find tap temp for each draw
taptemp=tap_temps(draw_count);
%Gather performance stats for solar system
if T_source(draw_count) < 60
    legionella_count=legionella_count+1;
end

if T_source(draw_count) < taptemp;
    warm_count=warm_count+1;
end
if T_ext(t) <= 0
    %Must turn gas heater's antifreeze system on,
    %which means it uses more electrical power
    gas_elec_cons=144;
else
    gas_elec_cons=62;
end

%power_inst_or_cont takes flow rate in m^3/s,
%but draw_rate is in
%L/s, so convert draw_rate to m^3/s:
flow_rate=actual_rate(draw_count)/1000;

energy_used_gas(draw_count)=actual_rate(draw_count)*...
    draw_rate_time(draw_count)*c_w*(T_heater-T_source...
    (draw_count))/real_eff_gas;
%Find power and cost required to heat water to the
%same temperature using
%an instantaneous electric system:
power_elec=HWSYS.power_inst_or_cont...
    (T_source(draw_count),T_heater,flow_rate,rho_w,...
    c_w,real_eff_elec);
energy_used_elec(draw_count)=power_elec*...
    draw_rate_time(draw_count);
energy_cost_gas(draw_count)=HWSYS.energy_running_cost...
    (['G',gas_elec_cons],energy_used_gas(draw_count),...
    draw_rate_time(draw_count));
energy_cost_elec(draw_count)=HWSYS.energy_running_cost...
    ('E',energy_used_elec(draw_count),draw_rate_time...
    (draw_count)); %AUD
CO2_emissions_gas(draw_count)=HWSYS.impact_running...
    (['G',gas_elec_cons],energy_used_gas(draw_count),...
    draw_rate_time(draw_count));
CO2_emissions_elec(draw_count)=HWSYS.impact_running...
    ('E',energy_used_elec(draw_count),draw_rate_time...
    (draw_count));

```

```

        if T_source(draw_count) > T_heater
            energy_used_elec(draw_count)=0;
            energy_used_gas(draw_count)=0;
            energy_cost_gas(draw_count)=0;
            CO2_emissions_gas(draw_count)=0;
            energy_cost_elec(draw_count)=0;
            CO2_emissions_elec(draw_count)=0;
        end
    end
end

energy_used_gas_tot=sum(energy_used_gas);
energy_used_elec_tot=sum(energy_used_elec);
CO2_emissions_elec_tot=sum(CO2_emissions_elec);
energy_cost_gas_tot=sum(energy_cost_gas);
CO2_emissions_gas_tot=sum(CO2_emissions_gas);
energy_cost_elec_tot=sum(energy_cost_elec);

if strcmp(circulating,'yes')== 1
    energy_used_gas_tot=energy_used_gas_tot+Q_circ;
    energy_cost_gas_tot=energy_cost_gas_tot+...
        HWSYS.energy_running_cost(['G',62],energy_used_gas_tot,...
        month_time-sum(draw_rate_time));
    CO2_emissions_gas_tot=CO2_emissions_gas_tot+...
        HWSYS.impact_running(['G',62],energy_used_gas_tot,...
        month_time-sum(draw_rate_time));
end

end

function [draw_rate,use_time,wait_time]=draw_info(type,...
    length_pipes,diam)

if strcmp(type,'L')== 1
    draw_rate=3.8/60; %kg/sec
    use_time=5; %sec
    LENGTH=length_pipes{1}{2};
elseif strcmp(type,'K')== 1
    %kg/sec; in my household, max flow rate is 12kg/min,
    %but typical rate will be lower
    draw_rate=5.7/60;
    total_use=5; %kg - based on experiment in my own household:
    %washing up a typical dinner for 4
    use_time=total_use/draw_rate;
    LENGTH=length_pipes{2}{2};
elseif strcmp(type,'S')== 1
    draw_rate=15/60; %kg/sec

```

```

        use_time=7*60; %sec
        LENGTH=length_pipes{3}{2};
    elseif strcmp(type,'B') == 1
        draw_rate=15/60; %kg/sec
        total_use=100; %kg, assuming 200L per bath
        use_time=total_use/draw_rate; %sec
        LENGTH=length_pipes{4}{2};
    elseif strcmp(type,'W') == 1
        draw_rate=7.5/60; %kg/sec
        total_use=50; %kg, assuming 5kg clothes washed
        use_time=total_use/draw_rate; %sec
        LENGTH=length_pipes{5}{2};
    elseif strcmp(type,'D') == 1
        draw_rate=5.7/60; %kg/sec
        total_use=13; %kg
        use_time=total_use/draw_rate; %sec
        LENGTH=length_pipes{6}{2};
    end
    wait_time=HWSYS.waiting_time(draw_rate/1000,diam,LENGTH);
end

end

end

```

N MATLAB run script for thermal model

```

clear all
clc
%%
addpath('/Users/bennettschneider/Documents/ANU/Semester 2, 2014/Systems/Portfolio/Mod
addpath('/Users/bennettschneider/Documents/ANU/Semester 2, 2014/Systems/Group project
import HWSYS.*
import GEA.*
%%
no_people=4;

if no_people==2
    volume_w=250; %L - volume of hot water tank
elseif no_people == 3
    volume_w=315;
elseif no_people >= 4
    volume_w=400;
end

months=1:12;

col_num=2; %Number of collectors

```

```

collector='evac';
mount_angle=35; %degrees
length_circuit=13.5; %m This is the length of pipework in circulating system
length_pipes={{'L',5},{'K',5},{'S',5},{'B',5},{'W',5},{'D',5}};

shower_length=7; %length of showers
%pipe diamters
diam_in=0.015; %m

%Length of hot pipe (3-4)
length34=0.5; %m
%Length of warm pipe (1-2)
length12=0.5; %m

%Find the power input of the collector:
tank_location='Outside';

%Specify the type of system
system='solar'; %'solar' or 'continuous'

circulating='no'; %'yes' or 'no'
circ_time=4; %circulating time per day (hours)

%time_24_hr_usage follows the following convention:
%{{time,'type',time_multiplier,rate_multiplier},...}
%Where time is the 24 hour decimal time, i.e. for 7:30pm, you would need to
%write 19.5. 'type' either refers to lavatory tap, kitchen tap, shower,
%bathtub, washing machine,or dishwasher usage, and are called respectively by:
% 'L', 'K', 'S', 'B', 'W', 'D'. Each of these will have a flow rate and typical
%time associated with them. The multiplier indicates when the use is more
%than one unit at a given time, e.g. if two people have a shower at the
%same time, or if there was more or less washing up to do than usual.

%default shower length is 7 mins, so modulate by:
mod_show=shower_length/7;
if no_people >= 6
    %Family of 6
    time_24hr_usage={{7,'S',mod_show*2,1},{8,'S',mod_show*1,1},{9,'K',1,1},{17,'K',1,1},
        {20,'K',1,1},{21.5,'S',mod_show*2,1},{22.5,'S',mod_show*1,1}}; %Daily draws
    %Specify non-daily draws over week
    time_weekly_usage={{12,'W',1,1},{12+24*2,'W',1,1},...
        {20.5+24,'D',1,1},{20.5+24*3,'D',1,1},{20.5+24*5,'D',1,1}};
elseif no_people == 5 || no_people == 4
    %Family of 4
    time_24hr_usage={{7,'S',mod_show*2,1},{9,'K',1,1},{17,'K',1,1},{20,'K',1,1},...
        {22,'S',mod_show*2,1}}; %Daily draws
    %Specify non-daily draws over week
    time_weekly_usage={{12,'W',1,1},{12+24*2,'W',1,1},...

```

```

        {20.5+24,'D',1,1},{20.5+24*4,'D',1,1},{20.5+24*6,'D',1,1}};
elseif no_people == 2
    %Family of 2
    time_24hr_usage={7,'S',mod_show*2,1},{9,'K',0.5,1},{20,'K',1,1}}; %Daily draws
    %Specify non-daily draws over week
    time_weekly_usage={{12,'W',0.5,1},{12+24*2,'W',0.5,1},...
        {20.5+24,'D',1,1},{20.5+24*4,'D',1,1}};
end

```

```

%Variables we do not necessarily want to change:
%assume HX fluid is similar to water (but has antifreeze properties)
c_HX=4184; %J/kgK
rho_HX=1000; %kg/m^3
%External pipes require minimum
%R-value of 1mK/W in climate region C (Standards Australia)
R_ext=1; %mK/W
%Internal pipes require minimum R-value of at least 0.3mK/W:
R_int=2;%0.3; %mK/W
%R-value of tank
R_tank=23*0.1761; %Converted from US to metric units: m^2K/W
%Temperature inside house
T_inside=20; %deg C
%Heat capacity of tank
rho_w=1000;
c_w=4184; %J/kgK
mass_w=volume_w; %kg
C_tank=mass_w*c_w;
real_eff_gas=0.74; %Energy conversion efficiency of continuous gas
%Resistance heaters have approximately perfect electricity to heat conversion
real_eff_elec=1;

%Suggested collector area for canberra region is 4m^2 (Standards Australia)
if strcmp(collector,'evac')== 1
    %For THERMANN evac tubes:
    %Aperture area of collectors:
    apert_A=col_numb*2.07; %m (22 tube system)

    eta0=0.687;
    a1=1.505;
    a2=0.0111;
    name='evacuated tube';
elseif strcmp(collector,'flat')== 1
    %For SunMaxx flat plate
    %Aperture area of collectors:
    apert_A=col_numb*1.874; %m

```

```

    eta0=0.801;
    a1=3.64;
    a2=0.0112;
    name='flat panel';
end

weeks=4;
days=weeks*7; %Number of days we run simulation over
runtime=3600*24*days; %Running time in seconds
timestep=30.001*60;%round(10*time_cu);
timesteps=runtime/timestep; %Number of timesteps
% random_set_temps_max=rand(1,days+1);
% random_set_temps_min=rand(1,days+1);
% random_set_irrad=rand(1,days+1);
% save('/Users/bennettschneider/Documents/ANU/Semester 2, 2014/Systems/Portfolio/Mode
% save('/Users/bennettschneider/Documents/ANU/Semester 2, 2014/Systems/Portfolio/Mode
% save('/Users/bennettschneider/Documents/ANU/Semester 2, 2014/Systems/Portfolio/Mode

time_168hr_usage=draw_times_general(time_24hr_usage,7); %Extend over a week
%Append these extra draws to the extended list of daily draws
start_append=length(time_168hr_usage)+1;
end_append=start_append+length(time_weekly_usage)-1;
time_168hr_usage(start_append:end_append)=time_weekly_usage;
%L/sec for any given timestep:
[weekly_draw_rate_L,weekly_water_cons,total_time_draw,weekly_actual_rate,...
    usage]=weekly_draw(time_168hr_usage,timestep,diam_in,length_pipes);
%Repeat this pattern over the number of weeks
draw_rate= repmat(weekly_draw_rate_L,1,weeks);
actual_rate=repmat(weekly_actual_rate,1,weeks);
draw_rate_time=repmat(total_time_draw,1,weeks);
%%

ann_warm_count=0;
ann_legionella_count=0;
Q_deliv_monthly=zeros(1,12);
Q_pipes_monthly=zeros(1,12);
Q_lost_draw_monthly=zeros(1,12);
Q_lost_cond_monthly=zeros(1,12);
energy_used_gas_monthly=zeros(1,12);
energy_used_elec_monthly=zeros(1,12);
energy_cost_gas_monthly=zeros(1,12);
energy_cost_elec_monthly=zeros(1,12);
CO2_emissions_gas_monthly=zeros(1,12);
CO2_emissions_elec_monthly=zeros(1,12);
T_heater_monthly=zeros(1,12);

for month=months;

```



```

%Calculate the various properties of solar hot water system as a function
%of time, as well as total heat transfers over the entire given runtime
[T_tank_draw,Q_deliv_tot,Q_pipe12_tot,Q_pipe34_tot,Q_lost_draw_tot,...
  Q_lost_cond_tot,T_fluid3,T_ext,T_tank,irradiance]=...
  solar(c_HX,rho_HX,R_ext,R_tank,c_w,C_tank,...
    apert_A,eta0,a1,a2,diam_in,length34,length12,days,...
    runtime,timestep,timesteps,month,mount_angle,...
    tank_location,draw_rate,collector);

%Total energy incident on collector in one month
Q_sol_monthly(month)=sum(irradiance*apert_A*timestep);
%Extend weekly usage cell over number of weeks given
time_usage= repmat(time_168hr_usage,1,weeks);
%Find required temprature for various draw types
tap_temps= tap_temp(time_usage);
temp_60_count=0;
temp_50_count=0;
for times=1:length(T_tank_draw)
  if T_tank_draw(times) > 60
    temp_60_count=temp_60_count+1;
    temp_50_count=temp_50_count+1;
  elseif T_tank_draw(times) > 50
    temp_50_count=temp_50_count+1;
  end
end

%We need above 60 deg C every week for 32 mins. Each timestep is 30
%mins, and each run is 4 weeks, so we need temp_60_count to be 5 or
%above if we want to set our booster to 50deg C.
%We need above 55 degrees for 6 hours every week for the booster to be
%set at 50 degrees. Thus, we require temp_50_count to be above 12*4=48

if strcmp(system,'solar')== 1
  T_source=T_tank_draw;
  if temp_50_count < 48 && temp_60_count < 5
%Set output temperature of non-renewable heaters according to Aust. Stand.
    T_heater=70;
  else
    T_heater=50;
  end
elseif strcmp(system,'continuous')== 1
  T_source=T_ext;
%Set output temperature of non-renewable heaters according to Aust. Stand.
  T_heater=50;
end
T_heater_monthly(month)=T_heater;
%Calculate the energy used, the cost, and the CO2 emissions associated with

```

```

%running continuous gas or instantaneous electric systems. Gas assumed to
%heat water centrally from outside the house, electric assumed to be
%decentralised, and heat at each draw point.
[energy_used_gas_month,energy_used_elec_month,energy_cost_gas_month,...
 energy_cost_elec_month,CO2_emissions_gas_month,...
 CO2_emissions_elec_month,warm_count,legionella_count]...
    =inst_gas_elec_boost(actual_rate,draw_rate,...
    draw_rate_time,T_ext,T_source,real_eff_gas,...
    real_eff_elec,T_heater,...
    tap_temps,rho_w,c_w,circulating,length_circuit,...
    diam_in,R_int,weeks,T_inside,circ_time);
Q_deliv_monthly(month)=sum(Q_deliv_tot);
Q_pipes_monthly(month)=sum(Q_pipe12_tot)+sum(Q_pipe34_tot);
Q_lost_draw_monthly(month)=sum(Q_lost_draw_tot);
Q_lost_cond_monthly(month)=sum(Q_lost_cond_tot);
energy_used_gas_monthly(month)=energy_used_gas_month;
energy_used_elec_monthly(month)=energy_used_elec_month;
energy_cost_gas_monthly(month)=energy_cost_gas_month;
energy_cost_elec_monthly(month)=energy_cost_elec_month;
CO2_emissions_gas_monthly(month)=CO2_emissions_gas_month;
CO2_emissions_elec_monthly(month)=CO2_emissions_elec_month;
ann_warm_count=ann_warm_count+warm_count;
ann_legionella_count=ann_legionella_count+legionella_count;
end

Q_sol_ann=sum(Q_sol_monthly);
Q_deliv_ann=sum(Q_deliv_monthly);
Q_pipes_ann=sum(Q_pipes_monthly);
Q_lost_draw_ann=sum(Q_lost_draw_monthly);
Q_lost_cond_ann=sum(Q_lost_cond_monthly);
energy_used_gas_ann=sum(energy_used_gas_monthly);
energy_used_elec_ann=sum(energy_used_elec_monthly);
energy_cost_gas_ann=sum(energy_cost_gas_monthly);
energy_cost_elec_ann=sum(energy_cost_elec_monthly);
CO2_emissions_gas_ann=sum(CO2_emissions_gas_monthly);
CO2_emissions_elec_ann=sum(CO2_emissions_elec_monthly);

%%
annual_water_cons=weekly_water_cons*52;
annotate={['Tank volume: ',...
num2str(volume_w),' L'],['System type: ',system],['Collector type: ',name],...
['Gas booster temp setting: ',num2str(T_heater),'deg C'],...
['Pipe diameter: ',num2str(diam_in*1000),'mm'],...
['Daily water consumption ',num2str(round(weekly_water_cons/7)),'L']};
if strcmp(system,'continuous') == 1
    if strcmp(circulating,'no') == 1
        title_suffix='for continuous heating systems';
    elseif strcmp(circulating,'yes') == 1

```

```

        title_suffix='for continuous circulating gas, and non-circulating electric';
    end
elseif strcmp(system,'solar')== 1
    title_suffix=['for ',name,' solar collector'];
end

%%
time=(0:timestep:runtime)/(24*3600);
% plot(time,ext_temp(T_max,T_min,time*3600))
figure;
plot(time,T_fluid3,time,T_ext(1:length(T_ext))...
    ,time,T_tank,time,irradiance/20,'LineWidth',3)
legend('Fluid 3','External T','Tank T','Normalised irradiance','Location','BestOutside')
%%
figure;
plot(time,draw_rate)
%%
figure;
bar([Q_sol_ann,Q_deliv_ann,-Q_pipes_ann,-Q_lost_draw_ann,-Q_lost_cond_ann]/10^9,'g')
set(gca,'XTickLabel',{'Sun','Collector','Circ. pipes','Water draw','Conduction'})
set(gca,'fontsize',28)
title(['Various annual losses associated with ',name,' solar collector'])
ylabel('Energy (GJ)')
annotation('textbox',[0.4,0.9,0,0],'String',annotate,'FontSize',12,...
    'Interpreter','tex','FitBoxToText','on','EdgeColor',[0 0 0])
%%
booster='gas';
%Plot sankey diagram of solar gain and losses in system
losses=[Q_sol_ann-Q_deliv_ann,Q_pipes_ann,Q_lost_cond_ann,Q_lost_draw_ann]/10^9;
inputs_actual=[Q_sol_ann]/10^9;
if strcmp(booster,'none')==1
    boosted=abs(10^9*(sum(losses)-sum(inputs_actual)))+10;
    input_add='Error';
elseif strcmp(booster,'gas')==1
    boosted=energy_used_gas_ann;
    input_add='Gas';
elseif strcmp(booster,'elec')==1
    boosted=energy_used_elec_ann;
    input_add='Elec';
end
inputs=[Q_sol_ann,boosted]/10^9;
unit='GJ';
labels={'Incident solar energy',input_add,'Heat lost: collector',...
    'Heat lost: pipes','Heat lost: tank','Heat extracted in water',input_add};
sep=[0,0];
drawSankey(inputs, losses, unit, labels,sep); %(Spelling, 2010)

annotation('textbox',[0.5,0.9,0,0],'String',{'Pipe length: ',...

```

```

    num2str(length34),' m'], ['Pipe R-value: ', num2str(R_ext), ' mK/W'], ...
    ['Mount angle: ', num2str(mount_angle), ' degrees'], ...
    ['Collector type: ', collector], ['Month: ', num2str(months)], ...
    ['Shower length :', num2str(shower_length), ' mins']], ...
    'FontSize', 12, 'Interpreter', 'tex', 'FitBoxToText', 'on', 'EdgeColor', [0 0 0])
%%
figure;
bar([energy_used_gas_ann, energy_used_elec_ann]/10^9, 'g')
set(gca, 'XTickLabel', {'Gas', 'Electric'})
set(gca, 'fontsize', 28)
title(['Non-renewable energy used annually ', title_suffix]);
ylabel('Energy (GJ)')
annotation('textbox', [0.4, 0.9, 0, 0], 'String', 'annotate', 'FontSize', 12, ...
    'Interpreter', 'tex', 'FitBoxToText', 'on', 'EdgeColor', [0 0 0])
%%
figure;
bar([energy_cost_gas_ann, energy_cost_elec_ann], 'g')
set(gca, 'XTickLabel', {'Gas', 'Electric'})
set(gca, 'fontsize', 28)
title(['Annual running cost for ', title_suffix])
ylabel('Cost (AUD$)')
annotation('textbox', [0.4, 0.9, 0, 0], 'String', 'annotate', 'FontSize', 12, ...
    'Interpreter', 'tex', 'FitBoxToText', 'on', 'EdgeColor', [0 0 0])
%%
figure;
bar([CO2_emissions_gas_ann, CO2_emissions_elec_ann], 'g')
set(gca, 'XTickLabel', {'Gas', 'Electric'})
set(gca, 'fontsize', 28)
title(['Annual effective CO2 emissions for ', title_suffix])
ylabel('CO2 emissions (kg)')
annotation('textbox', [0.4, 0.9, 0, 0], 'String', 'annotate', 'FontSize', 12, ...
    'Interpreter', 'tex', 'FitBoxToText', 'on', 'EdgeColor', [0 0 0])
%%
%Plots waiting times for relevant draw points, based on flow rates and
%distance of draw points from heat source
figure;
types=['L', 'K', 'S'];
%length_pipes={{'L', 5}, {'K', 4}, {'S', 5}, {'B', 5}, {'W', 5}, {'D', 5}};
%length_pipes={{'L', 15}, {'K', 10}, {'S', 15}, {'B', 15}, {'W', 5}, {'D', 5}};
%length_pipes={{'L', 0.5}, {'K', 0.2}, {'S', 2}, {'B', 15}, {'W', 5}, {'D', 5}};
length_pipes={{'L', 1.5}, {'K', 4}, {'S', 1.5}, {'B', 15}, {'W', 5}, {'D', 5}};
wait_time=zeros(1, length(types));
for type=1:length(types)
    [draw_rate, use_time, wait_time(type)]=draw_info(types(type), ...
        length_pipes, diam_in);
end
bar(wait_time)
type_names={'Lavatory', 'Kitchen', 'Shower'};

```

```

set(gca,'XTickLabel',type_names)
set(gca,'fontsize',28)
title(['Waiting times for various draw points'])
ylabel('Waiting time (seconds)')
%%
figure;
bar(usage)
draw_names={'Taps','Bath/Shower','Washing machine','Dishwasher'};
set(gca,'XTickLabel',draw_names)
set(gca,'fontsize',23)
title(['Typical weekly levels of hot water usage for a 4-person household'])
ylabel('Volume (L)')

```

O MATLAB source code for hot water survey results processing

```

classdef HWS
    methods (Static = true)

        function survey_list=get_results
            survey_raw=importdata('coding_hot_water_survey_raw.xls');
            survey_list=survey_raw.data;
        end

        function [property_count,index]=identify_property(survey_list,...
            property_index)
            %Takes the raw data in survey list and counts the number of
            %results in that column, e.g. if property_index = 3, this
            %function will output the number of people who have solar hot
            %water, and a list containing the survey number of those people
            size_survey=size(survey_list);
            property_count=0;
            index=[0];
            for row=1:size_survey(1)
                if survey_list(row,property_index) == 1
                    property_count=property_count+1;
                    if property_count > 0
                        index(property_count)=row;
                    end
                end
            end
        end

        function property_index=find_property_index(property)
            %This function takes a string (response) corresponding to a
            %specific response in the survey, e.g. '1_Gas' refers to

```

```

%whether or not people have a gas system, which is what Q1
%asks. The output of this is the column index of this response,
%e.g. for '1_Solar' this function will output the number 3.
property_list={'Survey #'},['Q1'],['1_Solar'],['1_Gas'],...
    ['1_Electric'],['1_Gas/electric'],['1_Dont know'],...
    ['1_Other'],['Q2'],['2_Continuous'],['2_Storage'],...
    ['2_Dont know'],['2_Other'],['Q3'],['3_Very S'],['3_S'],...
    ['3_Un S'],['3_Very Un S'],['3_Unsure'],...
    ['3_0'],['3_10'],['3_20'],['3_30'],['3_40'],['3_50'],...
    ['3_60'],['3_70'],['3_80'],['3_90'],['3_100'],['Q4'],...
    ['4_Very S'],['4_S'],['4_Un S'],['4_Very Un S'],['4_Unsure'],...
    ['4_0'],['4_10'],['4_20'],['4_30'],['4_40'],['4_50'],...
    ['4_60'],['4_70'],['4_80'],['4_90'],['4_100'],['Q5'],...
    ['5_Never'],['5_>3/week'],['5_1/week'],['5_1/month'],...
    ['Q6'],['6_Never'],['6_1/week'],['6_1/month'],['6_1/3 months'],...
    ['6_1/year'],['Q7'],['7_Evac'],['7_Flat'],['7_Parab'],...
    ['7_Dont know'],['7_Other'],['Q8'],['8_Gas'],['8_Electric'],...
    ['8_Gas/Electric'],['8_No'],...
    ['8_Dont know'],['8_Other'],['Q9'],['9_Roof'],['9_Indoors'],...
    ['9_Ground'],['9_Dont know'],['Q10'],['10_Apartment'],...
    ['10_House'],['Q11'],['11_1'],['11_2'],['11_3'],['11_4'],...
    ['11_5'],['11_6'],['11_7'],['11_8'],['11_9'],['Q12'],...
    ['12_Never'],['12_2/day'],['12_1/day'],['12_3/week'],...
    ['12_2/week'],['12_>1/week']};
for index=1:length(property_list)
    if strcmp(property_list(index),property) == 1
        property_index=index;
        break
    end
end
end

function count=count_responses(start,finish,surveys)
    %This function is designed to find the frequencies of each
    %possible response to a given question, under a given set of
    %surveys
    index_start=HWS.find_property_index(start);
    index_finish=HWS.find_property_index(finish);
    %Add up each response
    index=index_start:index_finish;
    count=zeros(1,length(index));
    for ii=1:length(index)
        count(ii)=HWS.identify_property(surveys,index(ii));
    end
end

function selected_surveys=indexed_surveys(survey_list,index)
    %This function compiles a subset of surveys defined by the index list

```

```

size_survey=size(survey_list);
selected_surveys=zeros(length(index),size_survey(2));
    for xx=1:length(index)
        if index == 0
            %If there are no results for a waiting period,
            %return an empty row
            selected_surveys(xx,:)=zeros(1,size_survey(2));
        else
            %For each waiting result, accumulate survey
            %responses in a submatrix
            selected_surveys(xx,:)=survey_list(index(xx),:);
        end
    end
end

function waiting_frequency_bar(frequencies,TITLE)
    figure;
    bar(frequencies)
    set(gca,'XTickLabel',{'0-4','5-14','15-24','25-34','35-44',...
        '45-54','55-64','65-74','75-84','85-94','>95'})
    set(gca,'fontsize',30)
    title(TITLE)
    ylabel('Frequency')
    xlabel('Seconds')

end

function average=sys_avg(count,value)
    %This function takes a vector containing the frequency of each
    %response (count), and pairwise multiplies this with the
    %corresponding value that each count corresponds to, thus
    %rapidly regenerating survey responses. It then adds them all
    %up and divides by the number of responses to find the average
    %value.
    average=sum(count.*value)/sum(count);
end

function data=generate_data(count,value)
    %This function generates an original data set from frequency
    %and value data by generating sets of values the size of the
    %frequency and assigning them to an empty data variable whose
    %size is equal to the size of the total data set
    data=zeros(1,sum(count));
    index_counter=0;
    for ii=1:length(count)
        index_counter=index_counter+count(ii);
        if ii == 1
            data(1:index_counter)=linspace(value(ii),value(ii),...

```

```

        count(ii));
    else
        data(index_counter-count(ii)+1:index_counter)=...
            linspace(value(ii),value(ii),count(ii));
    end
end
end

function standard_deviation=stnd(data,data_average)
    sum_of_differences_squared=0; %Start count
    for i=1:length(data) %Iterate for all elements
        sum_of_differences_squared=sum_of_differences_squared+...
            (data(i)-data_average)^2; %Accumulate a sum
    end
    standard_deviation=sqrt(sum_of_differences_squared/length(data)); ...
        %Apply formula directly
end

%Compute the average error for a set of data
function average_error=avgerr(standard_deviation,data)
    average_error=standard_deviation/(sqrt(length(data)-1)); ...
        %Apply formula directly
end

function question_ranges=find_Q_ranges(question)
    %This function takes a specific question number string as input
    %and outputs the range of index names whose columns contain the
    %responses for that question
    if strcmp(question,'Q1')== 1
        question_ranges={['1_Solar'],['1_Other']};
    elseif strcmp(question,'Q2')== 1
        question_ranges={['2_Continuous'],['2_Other']};
    elseif strcmp(question,'Q3_lik')== 1
        question_ranges={['3_Very S'],['3_Unsure']};
    elseif strcmp(question,'Q3')== 1
        question_ranges={['3_0'],['3_100']};
    elseif strcmp(question,'Q4_lik')== 1
        question_ranges={['4_Very S'],['4_Unsure']};
    elseif strcmp(question,'Q4')== 1
        question_ranges={['4_0'],['4_100']};
    elseif strcmp(question,'Q5')== 1
        question_ranges={['5_Never'],['5_1/month']};
    elseif strcmp(question,'Q6')== 1
        question_ranges={['6_Never'],['6_1/year']};
    elseif strcmp(question,'Q7')== 1
        question_ranges={['7_Evac'],['7_Other']};
    elseif strcmp(question,'Q8')== 1
        question_ranges={['8_Gas'],['8_Other']};
end

```



```

elseif strcmp(question,'Q9')== 1
    question_ranges={['9_Roof'],'9_Dont know'];
elseif strcmp(question,'Q10')== 1
    question_ranges={['10_Apartment'],'10_House'];
elseif strcmp(question,'Q11')== 1
    question_ranges={['11_1'],'11_9'];
end
end

function notional_error=notionerr(percentage_error,base_error,data)
%This function computes the average error associated with a
%list of data that has a notional percentage uncertainty and
%a notional flat uncertainty. It will take the notional error
%to be whichever error is higher
error=zeros(1,length(data));
for ii=1:length(data)
    if data(ii) == 0
        error(ii)=base_error;
    else
        error(ii)=percentage_error*data(ii);
    end
end
notional_error_1=sum(error)/length(error);

if notional_error_1 > base_error
    notional_error=notional_error_1;
else
    notional_error=base_error;
end
end

function [avg,stand_dev,error]=compare(value_list,names_list,...
    index_list,survey_list,questions,TITLE,YLABEL,...
    percentage_error,base_error,error_method)
%This function takes any list of systems or properties and
%compares the numerical averages of response with respect to
%one question or several related questions.
%e.g. it can compare the average waiting times for different
%system types, or the annual hot water loss rate for different
%systems and/or different household populations

%Value list must be a vector
%Names list must be a cell of string elements
%Index list must be a vector of cells, each containing one
%string element
%Questions my either by a cell of vectors or a vector of cells,
%each containing one string element
avg=zeros(1,length(index_list));

```

```

stand_dev=zeros(1,length(index_list));
error=zeros(1,length(index_list));
for ii=1:length(index_list)
    index=HWS.find_property_index(index_list(ii));
    [blank,index_system]=HWS.identify_property(survey_list,index);
    %Subset of survey responses
    surveys=HWS.indexed_surveys(survey_list,index_system);
    %If we are summing up the results from more than one
    %question, must find the ranges, and count the responses
    %for each question, then sum the results
    for xx=1:length(questions)
        question_range=HWS.find_Q_ranges(questions(xx));
        count(xx,:)=HWS.count_responses(question_range(1),...
            question_range(2),surveys);
    end
    size_system_count=size(count);
    if size_system_count(1) > 1
        count=sum(count);
    end
    avg(ii)=HWS.sys_avg(count,value_list);
    data=HWS.generate_data(count,value_list);
    stand_dev(ii)=HWS.stnd(data,avg(ii));
    if length(data) < 2
        avg_error=0;
    else
        avg_error=HWS.avgerr(stand_dev(ii),data);
    end
    %Choose whether percentage or flat error is more appropriate
    if strcmp(error_method,'percentage')== 1
        %Calculate average notional error based on a
        %percentage error
        notional_error=HWS.notionerr(percentage_error,...
            base_error,data);
    elseif strcmp(error_method,'flat')== 1
        notional_error=base_error;
    end
    if avg_error < notional_error
        %This will be caused by either coincidence or a too
        %small sample size
        error(ii)=notional_error;
    else
        error(ii)=avg_error;
    end
end
figure;
HWS.barwitherr(error,avg,'g')

```

```

        set(gca,'XTickLabel',names_list)
        set(gca,'fontsize',30)
        title(TITLE)
        ylabel(YLABEL)
    end

end

end
end

```

P MATLAB code for survey data processing

```

clc
clear all
%%
addpath('/Users/bennettschneider/Documents/ANU/Semester 2, 2014/Systems/Portfolio/Sur
addpath('/Users/bennettschneider/Documents/ANU/Semester 2, 2014/Systems/Portfolio/Sur
import HWS.*
survey_list=get_results;
size_survey=size(survey_list);
%%
%What is the most acceptable waiting time?
%Find all the waiting times for tap water, and all the waiting times for
%shower water and graph them separately
%Shower water pertains to Q3:
Q3_count=count_responses('3_0','3_100',survey_list);
%%
%Histogram of waiting times for hot shower
waiting_frequency_bar(Q3_count,'Responses for average length of time required to wait
%%
%Tap water pertains to Q4:
Q4_count=count_responses('4_0','4_100',survey_list);
%%
%Histogram of waiting times for hot tap
waiting_frequency_bar(Q4_count,'Responses for average length of time required to wait
%%
%Histogram of all waiting times for all systems
waiting_frequency_bar(Q3_count+Q4_count,'Responses for average length of time require
%%
%What is the most acceptable shower waiting time for all systems?
property_index_list=[{'3_0'},{'3_10'},{'3_20'},{'3_30'},{'3_40'},{'3_50'},...
    {'3_60'},{'3_70'},{'3_80'},{'3_90'},{'3_100'}];
start_lik=find_property_index('3_Very S');
finish_lik=find_property_index('3_Unsure');
%This will be a matrix whose columns contain the frequency of each likert
%response for each waiting time, which are the rows
Q3_lik_count=zeros(length(property_index_list),length(start_lik:finish_lik));
for ii=1:length(property_index_list)

```

```

    property_index=find_property_index(property_index_list(ii));
    [count_shower,index_shower]=identify_property(survey_list,property_index);
    %Create a new survey matrix containing only the surveys of the numbers
    %contained in the index_shower list
    surveys=indexed_surveys(survey_list,index_shower);
    Q3_lik_count(ii,:)=count_responses('3_Very S','3_Unable',surveys);
end
%%
%Histogram of likert responses for shower
waiting_times={['0-4'],['5-14'],['15-24'],['25-34'],['35-44'],['45-54'],...
    ['55-64'],['65-74'],['75-84'],['85-94'],['>95']];
for ii=1:4:length(property_index_list)
figure;
bar(Q3_lik_count(ii,:))
set(gca,'XTickLabel',{'V. Satisfied','Satisfied','Unsatisfied',...
    'V. Unsatisfied','Unable'})
set(gca,'fontsize',28)
title(['Linkert Responses for hot shower waiting time ',waiting_times{ii},' s'])
ylabel('Frequency')
end
%%
%What is the most acceptable tap waiting time for all systems?
property_index_list={['4_0'},{'4_10'},{'4_20'},{'4_40'},{'4_40'},{'4_50'}...
    ,{'4_60'},{'4_70'},{'4_80'},{'4_90'},{'4_100'}];
start_lik=find_property_index('4_Very S');
finish_lik=find_property_index('4_Unable');
%This will be a matrix whose columns contain the frequency of each likert
%response for each waiting time, which are the rows
Q4_lik_count=zeros(length(property_index_list),length(start_lik:finish_lik));
for ii=1:length(property_index_list)
    property_index=find_property_index(property_index_list(ii));
    [count_tap,index_tap]=identify_property(survey_list,property_index);
    %Create a new survey matrix containing only the surveys of the numbers
    %contained in the index_shower list
    surveys=indexed_surveys(survey_list,index_tap);
    Q4_lik_count(ii,:)=count_responses('4_Very S','4_Unable',surveys);
end
%%
%Histogram of likert responses for shower
waiting_times={['0-4'],['5-14'],['15-24'],['25-34'],['35-44'],['45-54'],...
    ['55-64'],['65-74'],['75-84'],['85-94'],['>95']];
for ii=1:3:length(property_index_list)
figure;
bar(Q4_lik_count(ii,:))
set(gca,'XTickLabel',{'V. Satisfied','Satisfied','Unsatisfied',...
    'V. Unsatisfied','Unable'})
set(gca,'fontsize',28)
title(['Linkert Responses for hot tap waiting time ',waiting_times{ii},' s'])

```

```

ylabel('Frequency')
end
%%
%Histogram of all likert responses
waiting_times={['0-4'], ['5-14'], ['15-24'], ['25-34'], ['35-44'], ['45-54'], ...
    ['55-64'], ['65-74'], ['75-84'], ['85-94'], ['>95']];
for ii=2;%1:3%length(property_index_list)
figure;
bar(Q4_lik_count(ii,:)+Q3_lik_count(ii,:))
set(gca,'XTickLabel',{'V. Satisfied','Satisfied','Unsatisfied',...
    'V. Unsatisfied','Unsure'})
set(gca,'fontsize',28)
title(['Linkert Responses for waiting time ',waiting_times{ii},' s'])
ylabel('Frequency')
end
%%
%Is there a type os system that is faster?
%Plot the spread of waiting times for gas, electric, storage, continuous
time_list=[2,10:10:100];
sys_names_list={'Solar','Gas','Electric','Storage','Continuous'};
system_index_list={['1_Solar'},{'1_Gas'},{'1_Electric'},{'2_Storage'},...
    {'2_Continuous'}];
questions={['Q3'], ['Q4']];
TITLE='Average waiting times for different types of systems, with error';
YLABEL='Waiting time (s)';
flat_time_error=5; %s
[sys_avg,sys_stand_dev,sys_error]=...
    compare(time_list,sys_names_list,system_index_list,survey_list,...
    questions,TITLE,YLABEL,0,flat_time_error,'flat');
%%
%How is reliability of a system related to water usage?
%Plot the average number of failures per year against number of people in
%household
failure_list=[0,52,12,4,1];
pop_names_list={'1','2','3','4','5','6','7','8','9'};
pop_index_list={['11_1'},{'11_2'},{'11_3'},{'11_4'},{'11_5'},{'11_6'},...
    {'11_7'},{'11_8'},{'11_9'}];
questions={['Q6']];
TITLE='Average number of annual failures for different household populations, with er
YLABEL='Number of failures (per year)';
percentage_fail_error=0.1;
flat_fail_error=1;
[pop_avg,pop_stand_dev,pop_error]=...
    compare(failure_list,pop_names_list,pop_index_list,survey_list,...
    questions,TITLE,YLABEL,percentage_fail_error,flat_fail_error,'percentage');
%%
%What is the most reliable system?
%Plot the average number of failures per year against number of people in

```

```

%household
failure_list=[0,52,12,4,1];
sys_names_list={'Solar','Gas','Electric','Storage','Continuous'};
system_index_list={['1_Solar'},{'1_Gas'},{'1_Electric'},{'2_Storage'},...
    {'2_Continuous'}];
questions={['Q6']};
TITLE='Average number of annual failures for different systems, with error';
YLABEL='Number of failures (per year)';
percentage_fail_error=0.1;
flat_fail_error=1;
[pop_avg,pop_stand_dev,pop_error]=...
    compare(failure_list,sys_names_list,system_index_list,survey_list,...
        questions,TITLE,YLABEL,percentage_fail_error,flat_fail_error,'percentage');
%%
%What is the most common system?
Q1_count=count_responses('1_Solar','1_Dont know',survey_list);
figure;
bar(Q1_count,'g')
set(gca,'XTickLabel',{'Solar','Gas','Electric','Gas or Electric','Unknown',...
    'Other'})
set(gca,'fontsize',28)
title('Prevalence of different hot water systems among ENGN2226 Students')
ylabel('Frequency')
%%
%What is the most common system?
Q1_count=count_responses('1_Solar','1_Dont know',survey_list);
figure;
bar(Q1_count,'g')
set(gca,'XTickLabel',{'Solar','Gas','Electric','Gas or Electric','Unknown',...
    'Other'})
set(gca,'fontsize',28)
title('Prevalence of different hot water systems among ENGN2226 Students')
ylabel('Frequency')
%%
%Compare prevalence of gas storage, gas continuous, electric storage,
%electric continuous
sys_names_list={'Gas','Electric'};
sys_index_list={['1_Gas'},{'1_Electric'}];
start_sys=find_property_index('2_Continuous');
finish_sys=find_property_index('2_Other');
sys_count=zeros(length(sys_index_list),length(start_sys:finish_sys));
for ii=1:length(sys_index_list)
    property_index=find_property_index(sys_index_list(ii));
    [count_sys,index_sys]=identify_property(survey_list,property_index);
    %Create a new survey matrix containing only the surveys of the numbers
    %contained in the index_sys list
    surveys=indexed_surveys(survey_list,index_sys);
    sys_count(ii,:)=count_responses('2_Continuous','2_Other',surveys);

```

```
end
%%
%Histogram of responses for gas and electric system types
figure;
bar([sys_count(1,1:2),sys_count(2,1:2),sys_count(1,3)+sys_count(2,3),...
     sys_count(1,4)+sys_count(2,4)], 'g')
set(gca, 'XTickLabel', {'Cont. Gas', 'Stor. Gas', 'Cont. Electric', ...
                       'Stor. electric', 'Unsure', 'Other'})
title('Prevalence of different types of electric and gas systems')
set(gca, 'fontsize', 28)
ylabel('Frequency')
%%
property_index_10=find_property_index('10_Apartment');
[property_count_10,index]=identify_property(survey_list,property_index_10);
```