Resurrecting Primex

TC-2012-03

Jack Devanney

Martingale, Inc

Sisyphus Beach

Tavernier, Florida

Date: 2012-03-28

This is a preliminary, incomplete draft for private discussion.
Please send any comments to djw1@c4tx.org.

Contents

1 Brief outline of Primex

Primex is a shell and tube heat exchanger (HX) design program written by ORNL about 1970.[1] It assumes a vertical counter-flow layout with tubes in the annulus between a central downcomer and the heat exchanger side.[2] The shell side fluid enters at the top flows to the bottom via the downcomer, then out to the tube bundle, and upwards through a set of disk and donut baffles, and out. The tubes enter at the top, proceed radially inward and then turn down. All the baffle spaces are assumed to be of the same length except the top most. This space must be large enough for four sixty degree tube bends, whose job is to avoid over-stress in the tubes due to differential thermal expansion. Figure 1 sketches the layout.

This drawing not finished.

Primex normally assumes the tube side fluid is heating the shell side, that is, the hot end is the top end. But it has the nominal capability of reversing everything. We have no evidence that this capability has been used or even debugged. So far we have not tried it ourselves.

Primex takes as input all the heat exchanger physical parameters except heat exchanger radius, baffle spacing, number of baffle spaces, and the radius of the expansion bends in the top of the heat exchanger. This input includes the diameter of the central downcomer, tube ID and OD, tube pitch, and the fraction of the annulus area occupied by the disks and the donuts.

Primex must also be given the heat load that must be transferred, the target hot and cold end temperatures of the two fluids, and the physical properties of the two fluids. From the fluid specific heats, the heat load, and the in/out temperatures, Primex can immediately determine the mass flow rates on both sides of the heat exchanger.

Finally, Primex must be given a max allowable pressure drop for each side of the heat exchanger.

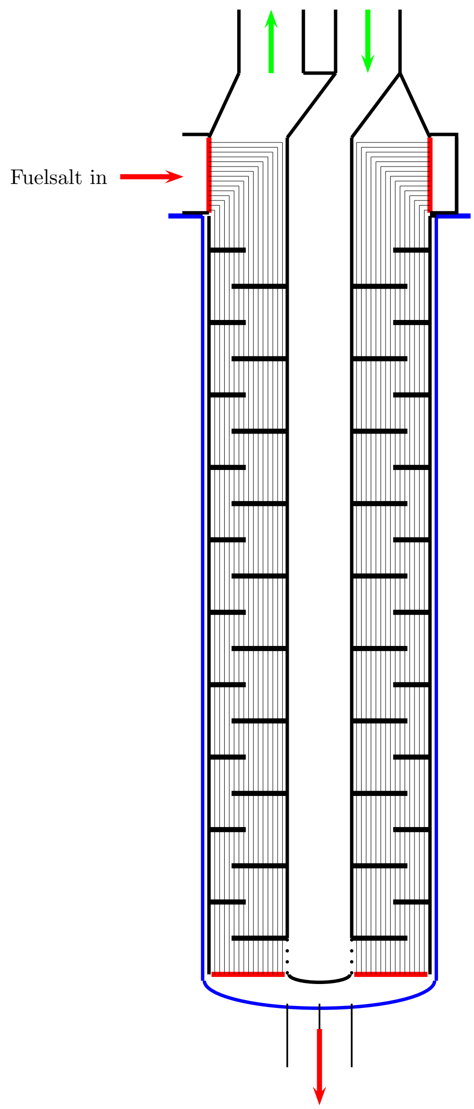
The core of Primex consists of six loops. Working from outermost to innermost, they are:

1. An outermost loop over baffle spacing, that is, the distance between each disk and the neighboring donut.
2. A loop over heat exchanger diameter. Upon entering this loop, the program determines the number of tubes and the fraction of the tubes in each of the three baffle space zones:
   (a) The inner zone from the central downcomer out to inside radius of the donut.
   (b) The middle zone from the inside radius of the donut to the outside radius of the disk.
   (c) The outer zone from the outside radius of the disk to the HX outer wall.
3. A stress check loop over the radius of the four expansion bends in the top of the HX.

[1] Bettis, C. et al, Computer Programs for MSBR Heat Exchangers, ORNL-TM-2815, June, 1971.

[2] Actually Primex does not assume vertical, but we will to make the description easier.

Figure 1: PHX Looking toward Pot



PHX OD: 1.742

PHX Top OD: 2.162

PHX HT: 7.320

Number of tubes: 5896

Tube length: 6.864

Tube OD (mm): 9.52

Tube wall thick (mm): 0.89

Baffle spacing: 0.285

Shell wall thick(mm) 12.50

Shield thick(mm) 31.25

Tube sheet thick(mm) 121.00

2012-03-26T14:55:00Z

4. A two pass adjustment loop. In the first pass, the number of baffle spaces required to cool the fluids down to the target cold temperatures is determined. In the second pass, a straight section of tube is inserted/deleted in/from the top baffle space to attempt to correct for any remaining difference between the target heat transferred and the calculated heat transferred.
5. A loop over number of baffle spaces. This loop starts at the hot (top) end of the heat exchanger, and keeps adding another baffle space until the fluids are down to the target cold temperatures. As it moves down the heat exchanger, it keeps track of the temperatures into and out of each baffle space, the corresponding heat transferred, and the pressure drops.
6. An innermost loop over temperature in each baffle space. Given the hot end temperatures at the top of the baffle space, this loop determines the temperatures at the cold end of the baffle space by trial and error. This requires determining the local heat transfer rates.

Upon leaving the adjustment loop, the program has a heat exchanger that is consistent with the heat load and the specified hot and cold end temperatures.

Primex then checks tube stress and adjusts the bend radius to reduce the stress if required. We have not yet implemented this logic.[3] In our current version, the bend radius is fixed by the user.

Upon leaving the stess check loop, we are at the bottom of the loop over heat exchanger diameter. At this point, Primex looks at the tube side pressure drop. It then proceeds by rather baroque logic to try and adjust the heat exchanger diameter so that the tube side pressure drop is close to but not above the max allowable. If it finds such a diameter, it drops out of the HX diameter loop.

Upon leaving the diameter loop, we are at the bottom of the loop over the baffle space length. At this point, Primex looks at the pressure drop on the shell side. It then proceeds by similar baroque logic to try and adjust the baffle space limit so that the shell side pressure drop is close to but not above the max allowable.

If all goes well, the result will be a heat exchanger which meets the heat load and cold/hot temperature requirements, avoids over-stressing the tubes, and ends up with pressure drops that are near but not above the user specified max allowable. This makes sense only if the target pressure drops are in fact close to the least cost pressure drops. Otherwise favoring a higher pressure drop over a lower would be nuts.

[3] The Primex stress analysis assumes the shell walls are at the shell fluid temperature. Since the shell fluid temperature on average is lower than the tube fluid, this sets up a compressive stress in the tubes. But in the 4541 PHX, the outer shell wall, the shield, will be heated by a layer of fuel salt on the outside. The average shield temperature may or may not be lower than the average tube temperature. The problem is the inner downcomer side. The inside of this material will be cooled by the incoming secondary salt. The average temperature of this material will certainly be lower than the average shell fluid temperature. It is far from obvious how accurate the Primex assumption is.

2 The Goal

Primex is a fiendishly clever program. But the coding is hideous. The program is written in 1960's style Fortran, with short, always cryptic and often meaningless variable names. There is nil indentation. Except for the tube stress analysis, there are no subroutines, just a single 500 line main program, with almost no comments. All variables are global. The loops are implemented with GOTO's. The order of the outermost loops seems strange. The result is some of worst spaghetti I have ever seen. Finally, the whole thing is in cursed English units.

Our goal was to totally restructure the code to make it far more flexible, and far more easily maintained. This involved separating the various functions into individual self-standing subroutines, giving the variables meaningful names, and getting rid of (almost all) the spaghetti. Table 1 shows a list of the subroutines. We also took advantage of this re-write to convert the program to strict SI units.

Table 1: List of Subroutines

Subroutine name	Function
primex_input	Read in and check input. Optionally print out input
primex_layout_tubes	Determine number of tubes, number of tubes in each baffle zone
primex_shell_fluxes	Determine the mass flux in each baffle zone
primex_shell_ht	Determine the baffle space shell side heat transfer rate
primex_tubes_ht	Determine the baffle space tube side heat transfer rate
primex_wall_ht	Determine the baffle space tube wall heat transfer rate
primex_shell_pd	Determine the baffle space shell side pressure drop
primex_tubes_pd	Determine the baffle space tube side pressure drop
primex_find_temp	Determine the cold end baffle space temperatures
primex_nbaffles	Determine the number of baffle spaces required
primex_output	Print results
primex	Mimic the Primex outer loops

In particular, the loop that determines the baffle space cold side temperatures is now its own subroutine, *primex_find_temp*, which in turns calls various heat transfer coefficient routines as necessary. Changing heat transfer or pressure drop correlations can now be done by changing subroutines, usually with no changes required in the calling routines.

The heat and soul of Primex is the number of baffle spaces loop. This too is now its own subroutine, *primex_nbaffles* which can be used in any number of ways, not just in the idiosyncratic Primex outer loops. For example, if one had a heat exchanger cost model, one could put this routine within loops that searched over tube diameter, heat exchanger diameter, and baffle spacing to determine the least cost heat exchanger meeting the requirements. This is precisely how we intend to use the core of the Primex logic.

Our main Primex program now only implements the outermost four loops, calling input, output and heat exchanger layout routines as necessary to set up each trial heat exchanger. The new main program contains less than 100 lines of executable code.

The rest of the code is in some 10 subroutines files which can be tested separately if desired. In fact, optional code to perform these tests in included in most of the subroutine files.

In order to process as rapidly as possible and make changes as easy as possible, our present version of Primex is written in Perl. But at this point translating the code to C or some other more structured language would not be a major undertaking. The key is the restructuring, not the language.

3 Bugs and Problems

The Primex restructuring exercise has revealed a number of problems with the Primex code, some of them quite serious. None of these errors appear to be corrected in the 1998 Ambidexter version of the code.

Tube-side enhancement factor Primex models the impact of helical grooves in the tube wall by an enhancement factor, *eh*, Both heat transfer rates and pressure drops are assumed to be linear in the enhancement factor. The enhancement factor is a function of local Reynolds number. In the 2581 notes, the tube side enhancement factor is capped at a Reynolds number of 10,000. But in the code there is no cap. In the sample problem, this over-states the tube side prssure drop at the hot end (Re = 11,000) of heat exchanger by up to 3%. However, the overall impact on pressure drop in the sample problem is not significant. The heat transfer error is similar, but in the opposite (improvement) direction.

Tube side friction factor In the 2581 notes, the tube-side friction factor uses different correlations for Reynolds numbers above and below 2100. But in the code, the high Reynolds number correlation is used regardless of Re. This did not effect the sample problem for which the tube side Reynolds number are always above 5000. But will generate serious errors if the program is used at lower tube side flow rates, for example, those generated by decay heat, natural circulation.

Tube side heat transfer rate The 2581 notes call for a different heat transfer correlation for $Re < 2100$, $2100 < Re < 12,000$, and $12,000 < Re$. corresponding to laminar, transitional, and fully turbulent regimes respectively. But in the code the fully turbulent heat transfer rate is used regardless of Reynolds number. The code in lines MSB 2450 to 2481 implementing the other two correlations is never reached. In the sample problem, the tube side Reynolds numbers range from about 10,000 down to 5,000. This result is a very serious over-statement of the tube side heat transfer rate, about 14% at the hot end and 24% at the cold end. If the program is used at low flow rates, the error skyrockets.

Shell side enhancement factor The 2581 notes call for the shell-side enhancement factor to be capped at 1.3 at Reynolds numbers above 10,000. But the code has no such cap. Since the shell side Reynolds numbers in the sample problem, range from 33,000 (hot end) to 17,000 (cold end), this result in a 25% over-statement of both the hot end shell side pressure drops and heat transfer rates. The overall effect on the sample 4541 heat exchange is about a 15% over-statement in both the shell side pressure drops and heat transfer rates. It is impossible to believe that ORNL was unaware of these differences.

Tube side pressure drop mystery To allow our Primex to mimic the original code, the user may set a switch *wants_primex_bugs* to yes. When we do that the individual heat transfer rates and pressure drops mimic Primex's 4541 sample numbers quite closely with one very big exception. Our tube side pressure drop at the hot end of 4541 heat exchanger is within 1% of Primex's. But as we move down the HX and the tube-side temperature drops, Primex's pressure drops decrease more rapidly than ours. At the cold end, the Primex pressure drop is 17% below the hot end. Our pressure drop is 2.5% below the hot end.[4] Detailed inspection of the change in density (increasing density decreases pressure drop), the change in the friction factor (increasing with increasing viscosity and decreasing Re), and the change in enhancement factor (decreasing with decreasing Re) reveals that 2.5% is the correct change. Since our code with *wants_primex_bugs* is supposedly a carbon-copy of the Primex code, ***this difference is currently unexplained***.[5] The overall impact on the sample problem is that, even if you run our code with *wants_primex_bugs* set, our total pressure drop on the tube side is roughly 10% above Primex's. When you run our Primex main program, this difference in pressure drop results in our code coming up with a rather different HX than the real Primex came up with.

[4] The decrease in pressure drop with decreasing temperature which means rapidly increasing viscosity is counter-intuitive. It results from the assumed decrease in the enhancement factor and the assumption that pressure drop is proportional to enhancement factor. How accurate these assumptions are is anybody's guess.

[5] The use of a strange set of English units (eg lb/hr) resulted in enormous (order of 10^9) conversion divisors. It is conceivable this caused numerical problems for 1970 vintage computers. I ran into such a problem in the 1960's which could have easily resulted in the loss of a submarine. Fortunately, the submarine's first dive was in shallow water and the result was a muddy bow. I was told Admiral Rickover was aboard.