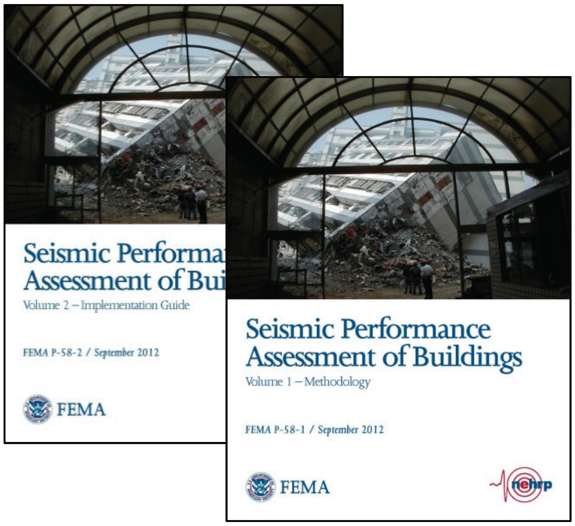

Background Document

FEMA P-58/BD-3.7.12

Risk Management Products Team Report: PACT 2 Verification Studies

Prepared by

Mahmoud M. Hachem
Degenkolb Engineers
235 Montgomery Street, Suite 500
San Francisco, California 94104

Submitted to

APPLIED TECHNOLOGY COUNCIL
201 Redwood Shores Parkway, Suite 240
Redwood City, California 94065
www.ATCouncil.org

Prepared for

FEDERAL EMERGENCY MANAGEMENT AGENCY
U.S. Department of Homeland Security
500 C Street, SW
Washington, D.C. 20472

March 29, 2013

Background Documentation

FEMA P-58 Background Documents are a series of reports documenting the technical background and source information for key aspects of the FEMA P-58 methodology and its implementation. These reports were developed over the course of the 10-year ATC-58/ATC-58-1 Projects funded under FEMA Contracts EMW-2001-RP-0056 and HSFEHQ-06-D-1105.

Background Documents were developed by consultants, serving at various levels within the project hierarchy, reporting the results of: (1) decisions on technical development protocols; (2) focused studies on the development of key aspects of the methodology; (3) documentation of recommended procedures; and (4) collection of available data for the development of structural and nonstructural fragilities. They were initially intended to serve as a record of the technical state-of-knowledge at the time they were produced, and as resources for the development of the eventual project reports. As such, they represent a snapshot in time, and may, or may not, match the technical content, recommended procedures, or data incorporated into the final methodology and its implementation.

This Background Document is intended for the purpose of providing supplemental knowledge to users of the FEMA P-58 methodology. Information contained herein has not been independently verified for accuracy as a stand-alone document, and may have been superseded in its final implementation within the methodology. Users of information in this document assume all liability arising from such use.

Notice

Cover illustration – Primary resource documents for the FEMA P-58 *Seismic Performance Assessment of Buildings, Methodology and Implementation* series of products: FEMA P-58-1, *Volume 1 – Methodology*, and FEMA P-58-2, *Volume 2 – Implementation Guide.*

Risk Management Products Team Report: PACT 2 Verification Studies

Performed by
Mahmoud M. Hachem, PhD, SE
Degenkolb Engineers

Final Report
September 29, 2012

Last Updated
March 29, 2013

Degenkolb Engineers
1300 Clay Street
SUITE 900
Oakland, California 94612-1455
510 272.9040 *phone*
272.9526 *fax*
www.degenkolb.com

Contents

1 Introduction

The computation engine of PACT was checked by performing extensive calculations that replicate all major calculations performed internally by PACT. All computations were performed independently and were only reliant on random numbers generated and reported by PACT. Hence, aside from random number generation, the review checked and replicated all quantities computed by PACT.

2 Data Checks

The PACT check started with the 3-story office building example that is installed with PACT. However, some modifications were made to the default file in order to introduce more complexity and test additional features such as setting some correlations and dispersions to non-zero values in order to test those functionalities. The data used for the checks, mainly the random numbers and associated data, are obtained from the data exported by PACT which is typically stored in one large XML data file. The data in the XML file is typically reported on a low level corresponding to a single or multiple quantities of a performance group (PG) under a combination of Analysis/Realization/Floor/Direction. In addition to checking results at this low level, additional checks were made at higher levels consisting of consequence results (loss, downtime, etc.) that are aggregated for complete floors, or the whole structure under a particular realization. These checks are important because of additional calculation steps performed during aggregation especially when computing collapse consequences or the possibility for red tags.

The low-level checking of the XML file results was typically performed as follows:

- Read contents of XML file line by line and import each PG quantity results into a separate row in an Excel spreadsheet. Realization damage and collapse damage were tracked separately in different sheets.
- For each PG quantity, replicate and check all PACT calculations (for realization damage and collapse) while relying only on the following
    - EDP values generated by PACT. Note that the statistical generation of the realizations was also independently checked.
    - Random values generated by PACT.
- Computed results are compared one by one to equivalent results reported by PACT for each PG.
- Computed results are aggregated and then compared to aggregated results exported by PACT, in order to check PACT results on a higher level.
- Checks are repeated for repair costs, downtime, fatalities, injuries, red tags, and long leads.

3 Checking of EDP calculations

The statistical generation of EDP's from the sample analyses was checked to confirm conformance with ATC 58 documentation (Appendix G). The Cholesky decomposition method was checked in the initial review. PACT later added the ability to use a second method, Eigenvalue Decomposition, which is more stable. This method was also checked.

3.1 Cholesky Decomposition

The following is a description of the Cholesky decomposition check.

PACT exports several files related to the Cholesky decomposition, for each intensity level and direction.

For each [EDPCholesky_NormalRandMatrix_IntX_DirY.csv] file (there are 8x2):

Use random variable (U) to compute EDP as follows:

1. Check the computation of matrices LY, MY, DY and RYY. The matrices are different for each Intensity and Direction combination.
2. Use above Matrices and U to compute EDP
3. Compare with [EDPCholesky_ILogMatrix_IntX_DirY.csv] and [EDPCholesky_FinalMatrix_IntX_DirY.csv]

All Cholesky calculations have been checked as follows:

1) Checked the Cholesky computations used to obtain the realization EDP's. Used the input (original) EDP's and followed the algorithm to compute all matrices for all realizations/8 analyses (time based analysis). All matrices were checked and confirmed. This completes the check of all exported Cholesky csv files:
   a) EDPCholesky_CholeskyMatrix_Int[1..8]_Dir[1..2].csv
   b) EDPCholesky_CorrCoefMatrix_Int[1..8]_Dir[1..2].csv
   c) EDPCholesky_FinalMatrix_Int[1..8]_Dir[1..2].csv
   d) EDPCholesky_ILogMatrix_Int[1..8]_Dir[1..2].csv
   e) EDPCholesky_InputMatrix_Int[1..8]_Dir[1..2].csv
   f) EDPCholesky_MeansMatrix_Int[1..8]_Dir[1..2].csv
   g) EDPCholesky_NormalRandMatrix_Int[1..8]_Dir[1..2].csv
   h) EDPCholesky_StdDevsMatrix_Int[1..8]_Dir[1..2].csv
2) Verified that the EDP's reported in the exported xml file are the same as those reported in the csv files, and both are consistent with the independently computed values.

3.2 Eigenvalue Decomposition

The following is a description of the Eigenvalue decomposition check.

PACT exports several files related to the Eigenvalue decomposition, for each intensity level and direction. The files are labeled with a number corresponding to the step number of the algorithm, as well as the name of the variable being output, which allows the results of the procedure to be checked at various steps in the algorithm.

There are 15 different files types starting from the original analysis EDP's to the final generated realization EDP's with 13 different intermediate results in between, including the generated random numbers that are used to calculate the EDP's of each realization.

The contents of each file was checked and compared to results computed using Matlab in compliance with the algorithm given in Appendix G of the FEMA P-58 report (page G-12, Volume 1, August 2012).

The results obtained from PACT matched the algorithm results independently computed using Matlab. This check was performed twice, once in absence of modeling dispersion ($\beta_m = 0$), and again with modeling dispersion (β_m) of 0.20, and both checks were satsifactory.

4 Checking of Damage State Calculations

Using exported XML data, computation were performed to verify that all damage and sub-damage states are computed correctly for:

1. Sequential fragilities
2. Simultaneous fragilities (see note below for slight difference between actual computation and PACT reference manual).
3. Mutually-exclusive fragilities

4.1 Damage State Checking Process

1. Select the correct EDP for the current realization and performance group from the EDP csv file "PGEDPValues_All.xlsx"
2. Import the fragility and consequence information from the XML results file (also checked fragility information in XML file against user interface).
3. Using the EDP and the fragility relationship corresponding to the PG, check the computation of the current damage state as follows:
    a. For sequential fragilities, check that the reported damage state is computed correctly.
    b. For simultaneous fragilities, check that all reported damage states are computed correctly. Note that random numbers are not currently reported in the XML data file for simultaneous damage states that are not active. Hence the calculations for those damage states could not be confirmed. This is currently a shortcoming in the exported XML file.

   c. For mutually-exclusive fragilities, check that the final reported damage state is correct.
4. Check the correct treatment of correlated vs. uncorrelated fragilities. For uncorrelated fragilities, each PG quantity is broken down into individual items with different possible damage states.

5 Checking of Cost/Time calculations

5.1 Checking of PG Quantities with Quantity Dispersion

More recent versions of PACT started allowing the quantity of a given PG to be defined using a lognormal probability distribution characterized by a mean and dispersion. Using the random numbers reported by PACT, the calculation of the random quantities was verified, and it was verified that those quantities were correctly used in subsequent consequence calculations.

5.2 Checking of *GlobalNumUnits* Value

Since consequence functions are generally dependent on the quantity of damaged items, the total quantity of a damaged item for the whole building is necessary in order to correctly compute the cost/time consequences. PACT reports this quantity as the **GlobalNumUnits** variable in the XML results file. For each fragility type, this represents the total quantity of items that are damaged in the building.

The GlobalNumUnits value was independently computed from the damage states for each realization and found to be consistent with the value reported with one caveat. PACT computes the GlobalNumUnits as the total of all damaged units in a given PG regardless of the damage state. This means that PG quantities for damage states 1, 2, 3, etc. of a given fragility are all added and used for the cost/time consequence functions for any of the damage states. An alternative interpretation of this parameter is to treat each damage state separately and to compute the quantities accordingly. This would result in lower quantities per damage state, and higher cost/time consequence values.

5.3 Checking of Cost calculations

5.3.1 Calculation of FinalCost (adjusted unit cost)

The first step in cost calculation is the computation of the unit repair cost. This is termed *FinalCost* in the PACT XML file. The calculation is performed as follows for each damaged performance group in a realization:

1. Based on the total repair quantity (see *GlobalNumUnits* above) for the PG, determine the mean (or median) unit cost of repair, *MeanCost* or *MedianCost*, depending on the type of consequence distribution (normal or lognormal). The mean cost value is unique to each damage state of each fragility, and can be determined by interpolating the

cost/quantity relationship defined in the fragility damage state consequence tab (in **Fragility Manager**).

2. Determine the dispersion or COV for the given damage state. This variable is also defined in the consequence tab of the **Fragility Manager**.
3. Using a randomly generated number (*DCRand*) that is generated by PACT and reported in the XML results file, compute the randomly generated unit cost. The computation is slightly different depending on whether a normal or lognormal distribution is assumed. :
   FinalCost = NormInv(*DCRand*,*MeanCost*,*MeanCost***COV*), for normal distribution
   FinalCost = LogInv(*DCRand*,*MedianCost*,*dispersion*), for lognormal distribution
4. The final value is adjusted to ensure that it is not negative
5. Note that median cost is obtained from the mean cost as follows:
   Median = Mean * Exp(-0.5*beta^2)

5.3.2 Calculation of TotalCost

For each Analysis/Realization/Floor/Direction/PG, compute *TotalCost* = *FinalCost***Quantity*. TotalCost values were compared with values in the [PGCosts_All.csv] file. The *Quantities* for each PG were obtained from Building Modeler.

5.3.3 Cost Aggregation

Finally, the costs are aggregated for each realization to report the total cost. For the case where no collapse occurs, this is simple summation of costs for all PG's under a realization. When collapse does occur, the total loss is equal to the replacement value of the building. This logic was checked against PACT exported results (file [RealCost_AllFloorsAllDirsAllPGs.csv]). The total repair cost is also set to the replacement cost when the residual displacement limit state is exceeded. While this aspect of the analysis was not included in the detailed review, it was confirmed by spot-checking instances of residual displacement exceedance.

5.4 Checking of Time calculations

5.4.1 Calculation of FinalTime (adjusted unit downtime)

Time calculations are performed similar to cost calculations as discussed above. The first step in time calculation is the computation of the unit repair time. This is termed *FinalTime* in the PACT XML file. The calculation is performed as follows for each damaged performance group in a realization:

1. Based on the total repair quantity (see *GlobalNumUnits* above) for the PG, determine the mean (median) unit time of repair, *MeanTime* or *MedianTime*, depending on the type of consequence distribution (normal or lognormal). The mean downtime value is unique to each damage state of each fragility, and can be determined by interpolating the time/quantity relationship defined in the fragility damage state consequence tab (in **Fragility Manager**).

2. Determine the dispersion or COV for the given damage state. This variable is also defined in the consequence tab of the **Fragility Manager**.
3. Using a randomly generated number (*DTRand*) that is generated by PACT and reported in the XML results file, compute the randomly generated unit time. The computation is slightly different depending on whether a normal or lognormal distribution is assumed. :
   FinalTime = NormInv(*DTRand*,*MeanTime*,*MeanTime***COV*), for normal distribution
   FinalTime = LogInv(*DTRand*,*MedianTime*,*dispersion*), for lognormal distribution
4. The final value is adjusted to ensure that it is not negative
5. Note that median downtime is obtained from the mean downtime as follows:
   Median = Mean * Exp(-0.5*beta^2)

5.4.2 Calculation of TotalTime

For each Analysis/Realization/Floor/Direction/PG, compute *TotalTime* = *FinalTime***Quantity*. TotalTime values were compared with values in the [PGTimes_All.csv] file. The calculated values are the same for each combination of Realization/Intensity/Direction/Floor/Fragility. These times are not yet adjusted for the number of maximum workers per floor, since they are at the PG level. This is later checked with aggregated down time results.

5.4.3 Downtime Aggregation

Finally, the downtimes are aggregated for each realization to report the total downtime. This is similar to aggregating cost with some additional steps. The main difference is that the downtimes for a floor are aggregated, then adjusted for the maximum number of workers on a floor as follows:

DownTime = TotalTime / (MaxWorkerPerSF*FloorArea),

For the case where no collapse occurs, the total downtime for the building is then computed as the sum of downtimes for each floor. When collapse does occur, the total downtime is automatically set equal to the replacement time of the building. This logic was checked against PACT exported results (file [RealTime_AllFloorsAllDirsAllPGs.csv]). The total downtime is also set to the replacement time when the residual displacement limit state is exceeded. While this aspect of the analysis was not included in the detailed review, it was confirmed by spot-checking instances of residual displacement exceedance.

Note: The computation of downtime as the serial sum of downtimes of the individual floors seems overly simplistic. This may be ok for low-rise buildings, but with multi-story buildings with distributed damage, it does not account for the possibility of performing repairs on multiple floors simultaneously. The PACT Technical Manual does refer to the possibility of performing a parallel repair scheme, but that option could not be located in the Building Modeler.

6 Checking of Deaths/Injuries Calculations

6.1 Checking of Death Calculations

For each PG quantity, computer the TotalDeaths under each PG for each realization (ignoring collapse), which involves a number of steps:

1. Using the randomly generated DeathRateBetaRand (should really be called DeathRateRand), and the mean death rate (DRate) and beta (DRateBeta) for the current fragility damage state, an adjusted DeathRate can be obtained as:
   AdjDeathRate = LOGINV(DeathRateBetaRand,LN(DRate), DRateBeta)
2. The DeathArea for the current PG quantity is obtained by multiplying by the affected area for this fragility/damage state, which is fragility property.
   DeathArea = AdjDeathRate * AffArea
3. For each Analysis/Realization/Floor/Direction/PG, compute
   TotalDeathArea = DeathArea*Quantity.
   The TotalDeathArea values were checked with values in the [PGFatalities_All.csv] file. The calculated values are checked against the reported csv values for each combination of Realization/Intensity/Direction/Floor/Fragility.
4. The **TotalDeathArea** numbers are compared to the PACT results exported to [PGFatalities_All.csv]. It should be noted that the exported [PGFatalities.csv] contains Total Death Area, and not number of deaths.
5. After the TotalDeathArea is computed and verified for each PG, the number is converted to an actual number of deaths (**TotDeaths**). This is done by multiplying by the floor's population per sq. ft. Note that the computation for PopPerSF, which is randomly generated, is described under the Collapse section (Section 0). The equation for TotDeaths is:
   TotDeaths = TotalDeathArea * PopPerSF

Once the TotDeaths due to realization damage (without collapse) are obtained for each PG, they can be aggregated to be compared to the aggregated values provided by PACT (for example in exported file [RealFatalities_AllFloorsAllDirsAllPGs.csv]). Since the final fatalities value reported by PACT includes collapse fatalities, those need to be considered as well, as follows:

1. For each Analysis/realization/floor, compute the number of fatalities per Algorithm 4.7.3 in the PACT Technical Manual. The algorithm returns the collapse deaths if there is collapse, otherwise, the deaths resulting from the TotalDeathArea above are used. The collapse deaths are also checked as discussed in a later section.
2. The total deaths are then added for each floor and direction to result in a total realization death count. This is compared to the results reported in the exported file "RealFatalities_AllFloorsAllDirsAllPGs.csv".

3. **Note**: The Collapse fatalities seem to always take precedence over damage induced fatalities, regardless of which one is larger. A better alternative might be to take the larger of the two. It would also be possible to include a portion of the deaths contributed by PG's in the uncollapsed area of a floor to the collapse deaths in case of collapse, in case the collapse area is less than the total area.

6.2 Checking of Injury Calculations

The injury calculations are performed similarly to death calculations. The same checks as above were performed, and all results were found to be correct.

7 Checking of Collapse Calculations

1. For each Analysis/Realization combination, compute **SaT1Final**, which is the realization demand Sa for collapse calculation, and compare it against the value reported in the xml results file. SaT1Final is computed using the following information:
    a. "Sa (T1) Median" and "Sa (T1) Dispersion" values that are defined in the Building Manager for each intensity.
    b. "SaT1Rand" random number for this realization reported in the XML file.
    c. SaT1Final = LOGINV(SaT1Rand,LN(SaT1Median),SaT1Beta)
    Note: In the latest PACT (version 2 beta, Jan 2012), it appears that SaT1Beta is always zero, and can no longer be defined in the user interface. Hence SaT1Final is now always equal to SaT1Median, and
2. For each Analysis/Realization combination, determine if there is collapse. First check the calculation of **CollapseEDPPercent**, which the probability of collapse for the current EDP value, using the following:
    a. SaCollapseMed and SaCollapseBeta values defining the collapse fragility as input in the Building Manger.
    b. SaT1Final as reported in the XML file and checked above.
    c. CollapseEDPPercent = LOGNORMDIST(SaT1Final, LN(SaCollapseMed) , SaCollapseBeta)
3. After checking CollapseEDPPercent above, check if there is **collapse**. This is done by comparing a random number (CollapseRand) with CollapseEDPPercent:
    a. If (CollapseRand<Collapse EDPPercent) then COLLAPSE
    b. CollapseRand is the collapse random variable for this realization as reported in the XML results file.
4. Check if the correct **CollapseMode** has been computed. This defines the applicable collapse mode and the corresponding collapsed area fractions on different floors. This calculation is done using:
    a. The random number CollapseModeRand reported in the XML results file.

   b. The defined collapse mode mutually exclusive probabilities as input in the Building Manager.
   c. The algorithm defined in PACT 2 Technical Manual (P2TM) in section 4.8.2.
   d. **Note**: The algorithm for determining collapse in counter-intuitive in that it considers the collapse modes in reverse order. However the results should be ok as long as the algorithm is used consistently. The same comment applies to determining the damage states for all fragilities.
   e. **Note**: The collapse mode number reported in the XML results file is zero based. This is confusing since a collapse mode of 0 is used both to defined an uncollapsed floor as well as a floor in collapse mode 1.
5. For each realization, check the **PopPerSF** variable. This variable is determined as follows:
   a. Peak number of occupants per 1000sf and dispersion as defined in the Building Manager (“Population” tab).
   b. Realization time of day, day of week and month
   c. PopRand random variable reported in XML results file.
   d. The sampled PopPerSF is computed as: SampPopPerSF=LOGINV(PopRand,LN(PeakOccupantPerSF),Beta)
   e. Two reduction factors Fday and Fmonth are computed to correct for daily and monthly variations of population density.
   f. PopPerSF = SampPopPerSF * Fday * Fmonth
   g. **Note:** The calculation of Fday does not seem to interpolate between hourly values. For example, realizations at 4:00pm, 4:20pm and 4:55pm are all assigned the population value at 4:00pm
   h. **Note**: The realization dates are generated between years 0 and 9999. This is not realistic and causes difficulties in checking the results since many software and programming libraries do not support dates before 1900 (e.g. Excel). Even if the appropriate libraries are used to handle those dates, calendars that old are not well defined. It is suggested that more realistic dates be used (e.g. 2000 to 2100), or to make the range customizable by the user. This issue is not expected to impact any of the results, but would make result checking easier.
6. Check the **CollapsePercent** variable. This variable defines the percentage of a floor that is collapsed under a given collapse mode, and can be obtained using a simple look-up. The CollapsePercent values are defined for each Collapse Mode on the “Collapse Fragility” tab of the Building Manager.
7. Check **CollapseDeaths** and **CollapseInjuries** for all collapsed realizations. These variables are stored in the XML data file. The calculation is performed by first obtaining a **CollapseDeathRate** and **CollapseInjuryRate** which define the rate of deaths and injuries among occupants in the collapsed areas. In order to obtain the final deaths and injuries

numbers, these rates are modified by PopPerSF * FloorArea * CollapsePercent. The calculations are performed as follows for CollapseDeaths. CollapseInjuries are computed similarly.

   a. CollapseDeathRate = LOGINV(*CollapseDeathRateRand*, Ln(*ModeFloorDeathRate*), *ModeFloorDeathBeta*), where *CollapseDeathRateRand* is a random variable between 0 and 1 that is reported in the XML results file. **Note** that since *ModeFloorDeathBeta* and *ModeFloorInjuryBeta* are both set to zero in the example project (for all modes and floors), this computation is not fully checked.
   b. ***CollapseDeaths*** = *CollapseDeathRate* * *PopPerSF* * *FloorArea* * *CollapsePercent*
   c. Note: that the previous version of PACT 2 did not properly consider *CollapsePercent* (Fraction of Floor Subjected to Collapse Debris) when computing CollapseDeaths. This issue appears to have been result in the current version.

8. The **CollapseDeaths** and **CollapseInjuries** was also checked for the case where the *CollapseDeathRate* and *CollapseInjuryRate* were defined probabilistically using a lognormal distribution. Those calculations were verified to work as expected.

8 Checking of Excessive Residual Displacement

Recent versions of PACT include a residual displacement feature that when triggered render a building unrepairable. This has significance on the calculation of several consequences such as repair time and unsafe placards. The residual displacement calculations were checked and the inclusion of the residual displacement damage state in consequence calculations was verified.

9 Checking of Unsafe Placard Counts

The Unsafe Placard calculations were checked and were found to be correct. Unsafe Placard results reported by PACT include whether a given simulation has an Unsafe Placard, and also provide a count of Unsafe Placard items for each PG, even for realizations that do not have an Unsafe Placard because the minimum number of unsafe items is not reached. This review confirmed that both the assignment of Unsafe Placard to the realization, as well as all the individual reported counts are computed correctly. It was also verified that PACT uses three separate criteria for generating an Unsafe Placard. These include (1) PG damage threshold, (2) Collapse damage state, and (3) excessive residual displacement damage state.

10 Checking of Long Lead Flag

PACT generated a long lead flag for items that require a long lead. A true or false flag is generated for each PG of each realization. All of the generated flags were checked for the sample cases, and were found to be correct.

11 General Comments

11.1 Numbering

Intensity, Realization, Direction and Floor numbers are all zero-based in the XML output file. These same numbers are one-based in the output CSV files. These values should be consistent. It is recommended that one-based numbers be used in both types of files.

11.2 Simultaneous Damage States

1. There is a difference in the way the random number is used compared to the Pact Reference Manual.
2. Only Active Damage States and their corresponding data are reported. Since random numbers for non-active states are not reported, they cannot be checked, and there is a still a chance that there are active states that are not being reported. Note however that all checked states appear correct, hence the algorithm used appears to be correct.

11.3 XML File Observations

1. Some tags appear as <TagState>0</TagState>, with 0 instead of Red or Green
2. <DeathRateBetaRand> should really be called <DeathRateRand>
3. <InjuryRateBetaRand> should really be called <InjuryRateRand>

12 Assumptions/Limitations

While the performed review is extensive and generally supports the overall validity of the computational engine, it is still possible that certain combinations of properties and options may still cause incorrect results. There are many combinations of settings that simply cannot be checked within the scope of this study. The latest update of this study greatly expanded the range of cases that are checked, included checking for most dispersion factors (e.g. for Collapse distributions, quantities, modeling uncertainty, and others.)

Other factors that were not checked in this scope, but that can easily be checked without the need to explore the low-level program output (XML results) or have been considered by other teams, include:

1. Effect of Region and Date cost multipliers (on *Project Info* tab)
2. Effect of Height, Hazmat, and Occupancy factors on cost
3. Effect of “Total Loss Threshold”
4. Graphical User Interface including:
    a. Examine Results window, including checking the binning logic
    b. Fragility Specification Manager
    c. Building Population Manager
    d. Reporting Tool
5. Simplified Analysis option

13 Summary of Findings/Recommendations

An extensive review of PATC version 2.0 was performed. The review focused on the verification of the internal computations of PACT. A careful item by item review was performed by replicating the majority of PACT calculations, using the internal data and random values generated and exported by PACT.

The review validated and confirmed the following calculations, at multiple levels of aggregation (PG, floor, realization):

- EDP Calculation (Cholesky decomposition method, and Eigenvalue Decomposition),
- Cost of repair,
- Downtime,
- Fatalities,
- Injuries,
- Unsafe Placards,
- Long Leads,
- Occurrence and effect of collapse on the above loss measures,
- Residual Displacement damage state, and effect on other consequences.

All of the checked PACT calculations were found to be correct, and their calculation was found to be consistent with the latest ATC-58 document, as generally consistent with the PACT technical manual.

Below is a compilation of some observations and comments that are mentioned in this report:

- Some random numbers are not exported by PACT. This mainly pertains to random numbers that are used to test for simultaneous damage states, where the random are only exported for damage states found to be active, which makes it impossible to check the calculations for inactive damage states.
- The Collapse fatalities/injuries seem to always take precedence over damage induced fatalities/injuries, regardless of which one is larger. Some cases were encountered where the damage induced injuries were higher than collapse injuries. This may be particularly significant for collapse modes that only affect a small portion of the building.
- The calculation of SaT1Final demand acceleration for collapse in the current PACT version does not involve using a beta, unlike the last version (circa 2010). It appears that SaT1Beta is always zero, and can no longer be defined in the user interface. Hence SaT1Final is now always equal to SaT1Median.
- The algorithm for determining collapse mode in counter-intuitive in that it considers the collapse modes in reverse order. However the results should be ok as long as the algorithm is used consistently. The same comment applies to determining the damage states for all fragilities.

- The collapse mode number reported in the XML results file is zero based. This is confusing since a collapse mode of 0 is used both to defined an uncollapsed floor as well as a floor in collapse mode 1.
- The realization dates are generated between years 0 and 9999. This is not realistic and causes difficulties in checking the results since many software and programming libraries do not support dates before 1900 (e.g. Excel).
- Numbering: Intensity, Realization, Direction and Floor numbers are all zero-based in the XML output file. These same numbers are one-based in the output CSV files. These values should be consistent. It is recommended that one-based numbers be used in both types of files