The Computation of the Mean First Passage Times for Markov Chains

Jeffrey J. Hunter[1]

Department Mathematical Sciences
School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology,
Private Bag 92006, Auckland 1142, New Zealand

ArXiv.com version 30 December 2016

Abstract

A survey of a variety of computational procedures for finding the mean first passage times in Markov chains is presented. The author recently developed a new accurate computational technique, an Extended GTH Procedure, Hunter (Special Matrices, 2016) similar to that developed by Kohlas (Zeit. fur Oper. Res., 1986). In addition, the author has recently developed a variety of new perturbation techniques for finding key properties of Markov chains including finding the mean first passage times, Hunter (Linear Algebra and its Applications, 2016). These recently developed procedures are compared with other procedures including the standard matrix inversion technique using the fundamental matrix (Kemeny and Snell, 1960), some simple generalized matrix inverse techniques developed by Hunter (Asia Pacific J. Oper. Res., 2007), and the FUND technique (with some modifications) of Heyman (SIAM J Matrix Anal. and Appl., 1995). *MatLab* is used to compute errors when the techniques are used on some test problems that have been used in the literature. A preference for the accurate procedure of the author is exhibited.

AMS classification: 15A09; 15B51; 60J10

Keywords: Markov chain; stochastic matrix, moments of first passage times, generalized matrix inverses

1. Introduction

In Markov chain (*MC*) theory mean first passage times (*MFPT*s) provide significant information regarding the short term behaviour of the *MC*. A review of *MFPT*s, together with details regarding stationary distributions and the group inverse of the Markovian kernel, is given in [18].
We refer the reader to this aforementioned article as it provides the relevant background to this paper and enables us to avoid repetition of the material. In Hunter [18], which

[1] *E-mail address: jeffrey.hunter@aut.ac.nz*

focuses on computational techniques for the key properties of irreducible *MC*s using perturbation techniques, we commented that in a sequel paper we would consider a variety of other techniques to get a better impression as to whether perturbation procedures may in fact prove to be suitable alternatives. We address these issues in this paper.

We firstly however set the scene by reintroducing the notation that was used in [18].

Let $\{X_n, n \geq 0\}$ be a finite *MC* with state space $S = \{1, 2, \ldots, m\}$ and transition matrix $P = [p_{ij}]$, where $p_{ij} = P\{X_n = j \mid X_{n-1} = i\}$ for all $i, j \in S$.

The stationary distribution $\{\pi_j\}$, $(1 \leq j \leq m)$, exists and is unique for all irreducible *MCs*, that $\pi_j > 0$ for all j, and satisfies the equations (the *stationary equations*)

$$\pi_j = \sum_{i=1}^{m} \pi_i p_{ij} \text{ with } \sum_{i=1}^{m} \pi_j = 1. \tag{1.1}$$

If $\boldsymbol{\pi}^T \equiv (\pi_1, \pi_2, \ldots, \pi_m)$, the *stationary probability vector*, and $\boldsymbol{e}$ is a column vector of 1's, the stationary equations (1.1) can be expressed as

$$\boldsymbol{\pi}^T (I - P) = \boldsymbol{0}^T, \text{ with } \boldsymbol{\pi}^T \boldsymbol{e} = 1. \tag{1.2}$$

Let $T_{ij} = \min\left[n \geq 1, X_n = j \mid X_0 = i\right]$ be the first passage time from state i to state j (first return when $i = j$) and define $m_{ij} = E\left[T_{ij} \mid X_0 = i\right]$ as the *MFPT* from state i to state j (or mean recurrence time of state i when $i = j$). For finite irreducible *MCs* all the m_{ij} are well defined and finite. Let $M = [m_{ij}]$ be the *MFPT* matrix. Let $\delta_{ij} = 1$, when $i = j$ and 0, when $i \neq j$. Let $M_d = \left[\delta_{ij} m_{ij}\right]$ be the diagonal matrix formed from the diagonal elements of M, and $E = [1]$ (i.e. all the elements are unity).

It is well known ([19]) that, for $1 \leq i, j \leq m$,

$$m_{ij} = 1 + \sum_{k \neq j} p_{ik} m_{kj}. \tag{1.3}$$

In particular, for all $j \in S$, the mean recurrence time of state j is given by

$$m_{jj} = 1/\pi_j. \tag{1.4}$$

From (1.3) and (1.4) it follows that M satisfies the matrix equation

$$(I - P)M = E - PM_d, \text{ with } M_d = \left(\Pi_d\right)^{-1}. \tag{1.5}$$

Note that the expression (1.5) typically involves knowledge of Π_d, i.e. the stationary probabilities. In this paper we are not focussing on the computation of stationary distributions but when we require such terms when they are not explicitly derived in carrying out the computations for M, we typically use the *GTH* algorithm of Grassman, Taksar and Heyman [3], (or equivalently the State Reduction procedure of Sheskin [23]), as these are known to give accurate results with no subtractions being involved. There are however other alternative procedures that could be used, for example the *eig* procedure of *MatLab*.

In this paper we provide twelve procedures for solving, in effect, equations (1.3) or (1.5), for the *MFPTs*. In Section 2 we give some direct procedures (Procedures 1 and 2) based upon utilising matrix inverses. In Section 3 we summarise the six perturbation procedures (Procedures 3 to 8), given by Hunter in [18]. In Section 4 we outline the extended *GTH* procedure (*EGTH*) of Hunter [17] based upon Kohlas [20] (Procedure 9), while in Section 5 we outline the *FUND* procedure of Heyman [7] which we modify to simpler procedures that enables us to find the *MFPTs* without directly computing the fundamental matrix, to yield Procedures 10, 11 and 12.

In the final section, utilising the test problems used initially by Harrod and Plemmons ([5]) in comparing different techniques for computing the stationary probabilities, we use *MatLab* computations to compare errors, typically the maximum absolute errors, the overall residual errors in our computations, in both single and double precision, and the number additional accurate digits achieved with double precision over single precision. This leads to the conclusion that the Hunter *EGTH* Procedure ([17]) generally gives us the most accurate results. Such a procedure is much more accurate than any of the perturbation procedures.

Section 2: Computation of *MFPTs* using matrix inverses

If A is an $m \times m$ matrix of real elements and X is any $m \times m$ matrix that satisfies the condition $AXA = A$, then X is said to be a one-condition generalised matrix inverse, a g-inverse, of A, and is often written as A^-. If A is non-singular then $A^- = A^{-1}$.

All g-inverses of $I - P$ can be expressed in terms utilising matrix inverses, as pioneered by Hunter [11]. The general result is as follows:

Theorem 2.1: *Let P be the transition matrix of a finite irreducible Markov chain with m states and stationary probability vector $\boldsymbol{\pi}^T = (\pi_1, \pi_2, \ldots, \pi_m)$. Let $\mathbf{e}^T = (1, 1, \ldots, 1)$ and $\mathbf{t}$ and $\mathbf{u}$ be any vectors.*

(a) $I - P + \mathbf{t}\mathbf{u}^T$ *is non-singular if and only if* $\boldsymbol{\pi}^T\mathbf{t} \neq 0$ *and* $\mathbf{u}^T\mathbf{e} \neq 0$.

(b) *If* $\boldsymbol{\pi}^T\mathbf{t} \neq 0$ *and* $\mathbf{u}^T\mathbf{e} \neq 0$ *then* $[I - P + \mathbf{t}\mathbf{u}^T]^{-1}$ *is a one-condition g-inverse of* $I - P$.

(c) *All one-condition g-inverses of $I - P$ can be expressed as*

$$A^- = [I - P + \mathbf{t}\mathbf{u}^T]^{-1} + \mathbf{e}\mathbf{f}^T + \mathbf{g}\boldsymbol{\pi}^T$$ *for arbitrary vectors $\mathbf{f}$ and $\mathbf{g}$.*

□

Well-known special g-inverses of $I - P$ are Kemeny and Snell's *fundamental matrix* $Z = [I - P + \Pi]^{-1}$ where $\Pi = \mathbf{e}\boldsymbol{\pi}^T$, introduced in [19], (and initially shown to be a g-inverse of $I - P$ by Hunter ([10])) and Meyer's *group inverse of* $I - P$ given by $A^\# = Z - \Pi$, ([21]). ($A^\#$, is more restrictive than a simple g-inverse in that it is the unique g-inverse that satisfies, $(I - P)A^\# = A^\#(I - P) = I - \mathbf{e}\boldsymbol{\pi}^T$, $A^\#\mathbf{e} = \mathbf{0}$ and $\boldsymbol{\pi}^T A^\# = \mathbf{0}^T$ where $A = I - P$.

The reason we introduce g-inverses is that one condition g-inverses of $I - P$ are typically used to solve systems of linear equations involving $I - P$ (as in (1.2) and (1.5)), and are hence called "equation solving" g-inverses.

Specialising to the equations of the form (1.5) to be solved for *MFPTs*, we have the following general result, (see [11],[12]):

Theorem 2.2: Let *A and B be given m × m matrices, and X is an unknown m × m matrix.*

A necessary and sufficient condition for $AX = B$ *to have a solution is that*

$AA^{-}B = B$. *If this consistency condition is satisfied the general solution is given by*

$X = A^{-}B + W - A^{-}AW$, *where W is an arbitrary matrix.* (2.1)

□

Solving the equations given by (1.5), using Theorems 2.1 and 2.2, yield the following general results for finding the *MFPTs* of *MCs* (see [11], [12] for (a) and [16] for (b) and (c)):

Theorem 2.3:

(*a) If G is any g-inverse of* $I - P$, *then the MFPT matrix M, is given as*

$$M = [G\Pi - E(G\Pi)_d + I - G + EG_d]D, \quad (2.2)$$

where $D = (\Pi_d)^{-1} = \left[(\boldsymbol{e}\boldsymbol{\pi}^T)_d\right]^{-1}$.

(b) If $H \equiv G(I - \Pi)$ *then H is a g-inverse of* $I - P$ *with* $H\boldsymbol{e} = \boldsymbol{0}$ *and*

$$M = [I - H + EH_d]D. \quad (2.3)$$

(c) $G\boldsymbol{e} = g\boldsymbol{e}$ *for some g if and only if*

$$M = [I - G + EG_d]D. \quad (2.4)$$

□

Special cases of (2.4) for M are $G = Z$ and $G = A^{\#}$.

Note that the implementation of Theorem 2.3 typically requires prior calculation of the stationary probability vector $\boldsymbol{\pi}^T$. It is well known that the most accurate procedure for finding the stationary probabilities is the *GTH* algorithm, ([3]). See Section 4 for the details.

Theorem 2.3 above leads to the following two procedures.

Procedure 1: (Standard method)

Given an irreducible P

(i) Compute the stationary probability vector $\boldsymbol{\pi}^T$.

(ii) Compute the fundamental matrix $Z = [I - P + \boldsymbol{e}\boldsymbol{\pi}^T]^{-1}$.

(iii) Compute $M = [I - Z + EZ_d][(\boldsymbol{e}\boldsymbol{\pi})_d]^{-1}$.

This is the original procedure developed by Kemeny and Snell ([19]) and has been universally used in the past. It, of course, suffers from the requirement to compute a matrix inverse that can lead to significant inaccuracy, (see (8], [18]). As identified above,

prior to computing Z, the stationary probability vector $\boldsymbol{\pi}^{\mathrm{T}}$ is required. We use the *GTH* algorithm to compute $\boldsymbol{\pi}^{\mathrm{T}}$.

Hunter ([15]) established a number of results regarding expressions for the *MFPTs* using a range of simple matrix inverses of the form given in Theorem 2.1, (typically with $\boldsymbol{f}$ and $\boldsymbol{g}$ taken as zero vectors.) The simplest result is given as follows.

Procedure 2: (Simple method)
Given an irreducible P
(i) Compute the g-inverse $G = \left[I - P + \boldsymbol{e}\boldsymbol{e}_b^T \right]^{-1}$ *where* $\boldsymbol{e}_b^T$ *is a vector with* 1 *in the b-th position and* 0 *elsewhere.*
(ii) Compute, $\boldsymbol{\pi}^T = G\boldsymbol{e}_b^T$ *so that if* $G = [\, g_{ij} \,]$ then $\pi_j = g_{bj},\ j = 1,2,\ ...,m.$
(iii) Compute $M = [I - G + EG_d\,][\,(\boldsymbol{e}\boldsymbol{\pi}^T)_d]^{-1}$.

Thus following one matrix inversion (actually only the *b-th* row, typically the first row, for the stationary distribution), one can find the stationary probabilities and the mean first passage times. The choice of b is arbitrary. We take $b = 1$ in our test examples (in Section 6).

As the above two procedures both require the evaluation of matrix inverses we do not expect them to perform well in examples where we have either a large number of states or ill-conditioned matrices. One way to expedite the calculations is to consider using a perturbation technique that in effect carries out the required computations sequentially, row by row. We consider that in the next section.

Section 3: Computation of *MFPTs* using perturbation procedures

The general idea behind the perturbation procedures, which are considered in detail in Hunter [18], is the following. Start with a simple transition matrix P_0 with known or easily computed stationary probability vector $\boldsymbol{\pi}_0^T$, mean first passage time matrix M_0 and a simple g-inverse G_0 or easily computed fundamental matrix Z_0 or group inverse $\mathrm{A}_0^{\#}$. Then sequentially change the transition matrix P_0 by replacing the *i-th* row of P_0 with the *i-th* row of the given transition matrix P (i.e. $\boldsymbol{p}_i^T = \boldsymbol{e}_i^T P$) ($i = 1, 2, \ldots, m$) to obtain P_i ending up with $P_m = P$. Thus, as in [18], let $P_0 = \sum_{i=1}^{m} \boldsymbol{e}_i \boldsymbol{p}_{(0)i}^T$ so that if $P = \sum_{i=1}^{m} \boldsymbol{e}_i \boldsymbol{p}_i^T$ then $P_i = P_{i-1} + \boldsymbol{e}_i \boldsymbol{b}_i^T$ with $\boldsymbol{b}_i^T = \boldsymbol{p}_i^T - \boldsymbol{p}_{(0)i}^T$, for $i = 1, 2, \ldots, m$. Thus we update $\boldsymbol{\pi}_{i-1}^T$, M_{i-1}, and G_{i-1} (or Z_{i-1}, $A_{i-1}^{\#}$) to $\boldsymbol{\pi}_i^T$, M_i, and G_i (or Z_i, $A_i^{\#}$) finishing with $\boldsymbol{\pi}^T = \boldsymbol{\pi}_m^T, M = M_m$ and $G = G_m$, (or $Z = Z_m$, $A^{\#} = A_m^{\#}$).
The simplest structure to start with is the irreducible transition matrix $P_0 = \boldsymbol{e}\boldsymbol{e}^T/m$. This also ensures that each subsequent updated transition matrix is also irreducible. Thus $\boldsymbol{p}_{0(i)}^T = \boldsymbol{e}^T/m$, $\boldsymbol{\pi}_0^T = \boldsymbol{e}^T/m$, $Z_0 = I$, $A_0^{\#} = I - \boldsymbol{e}\boldsymbol{e}^T/m$, and $M_0 = m\boldsymbol{e}\boldsymbol{e}^T$.

The successive updates effectively make use of the Sherman- Morrison (1949) formula, ([22]), for computing matrix inverses:

If A is invertible, then $(A+\boldsymbol{u}\boldsymbol{v}^T)^{-1} = A^{-1} - \dfrac{1}{1+\boldsymbol{v}^T A^{-1}\boldsymbol{u}} A^{-1}\boldsymbol{u}\boldsymbol{v}^T A^{-1}$.

In particular, $[I-\boldsymbol{a}\boldsymbol{h}^T]^{-1} = I + (1/1-\boldsymbol{h}^T\boldsymbol{a})\boldsymbol{a}\boldsymbol{h}^T$, when $\boldsymbol{h}^T\boldsymbol{a} \neq 1$.

We do not give the details as the results are derived in [18].

The first perturbation procedure was an extension to the procedure of Hunter (1991) ([14]) updating a one-condition generalized inverse to find successive stationary probability vectors, leading to computation of the *MFPT* matrix. Let $G_i = [I - P_i + \boldsymbol{t}_i\boldsymbol{u}_i^T]^{-1}$. We update the g-inverse G_{i-1} to G_i successively with $\boldsymbol{t}_0 = \boldsymbol{e}$, $\boldsymbol{u}_0^T = \boldsymbol{e}^T/m$ with $G_0 = [I - P_0 + \boldsymbol{t}_0\boldsymbol{u}_0^T]^{-1} = I$. We use Theorem 2.3(b) above as this eliminates the requirement to find the group inverse but utilises the structure of H, a particular g-inverse of $I - P$. This is Algorithm 1 in [18].

Procedure 3: (G-inverse update – Pert AL1)

(i) Let $G_0 = I$, $\boldsymbol{u}_0^T = \boldsymbol{e}^T/m$.

(ii) For $i = 1, 2, ..., m$, let $\boldsymbol{p}_i^T = \boldsymbol{e}_i^T P$, $\boldsymbol{u}_i^T = \boldsymbol{u}_{i-1}^T + \boldsymbol{p}_i^T - \boldsymbol{e}^T/m$,

$G_i = G_{i-1} + G_{i-1}(\boldsymbol{e}_{i-1} - \boldsymbol{e}_i)(\boldsymbol{u}_{i-1}^T G_{i-1}/\boldsymbol{u}_{i-1}^T G_{i-1}\boldsymbol{e}_i)$.

(iii) At $i = m$, let $G_m = G$ and $\boldsymbol{\pi}^T = \boldsymbol{\pi}_m^T = \dfrac{\boldsymbol{u}_m^T G_m}{\boldsymbol{u}_m^T G_m \boldsymbol{e}}$.

(iv) Compute $H = G(I - \boldsymbol{e}\boldsymbol{\pi}^T)$.

(vi) Compute $M = [I - H + E(diag(H))]D$ where

$E = [1]$ *and* $D = inv[diag(\boldsymbol{e}\boldsymbol{\pi}^T)]$.

For the next procedure we consider an extension to Procedure 3 through updating using matrix procedures that yield, in tandem at each step, the stationary probability vectors and the group inverses. This is Algorithm 2 in [18].

Procedure 4: (Group inverse update – Pert AL2)

Start with P.

(i) Set $R_0 = I - \boldsymbol{e}\boldsymbol{e}^T/m$.

(ii) For $i = 1, 2, \ldots, m$, let $\boldsymbol{p}_i^T = \boldsymbol{e}_i^T P$, $\boldsymbol{b}_i^T = \boldsymbol{p}_i^T - \boldsymbol{e}^T/m$,

$R_i = R_{i-1} + \dfrac{1}{1 - \boldsymbol{b}_i^T R_{i-1}\boldsymbol{e}_i} R_{i-1}\boldsymbol{e}_i\boldsymbol{b}_i^T R_{i-1}$.

(iii) Compute $\boldsymbol{\pi}^T = \boldsymbol{e}_1^T - \boldsymbol{e}_1^T(I - P)R_m$.

(iv) Compute $M = [I - R_m + E(diag(R_m))]D$, where $E = [1]$ and $D = inv[diag(\boldsymbol{e}\boldsymbol{\pi}^T)]$.

Some simplifications to this algorithm are
possible, as not all the calculations are required. In (*ii*) note that $R_i = R_{i-1}\left(I + C_i\right)$, where $C_i = \left(1/k_i\right)\boldsymbol{e}_i\boldsymbol{b}_i^T R_{i-1}$ and $k_i = 1 - \boldsymbol{b}_i^T R_{i-1}\boldsymbol{e}_i$ implying that C_i has all terms zero except terms in the *ith* row. So that at the *ith* recursion the only terms that are updated are in the first *i* rows with the rows numbered *i*+1, *i*+2, …, *m* remaining unchanged.

Rather than carry out the updating by stationary probability vectors we can use a procedure based on updating the limiting matrix $\Pi = \boldsymbol{e}\boldsymbol{\pi}^T$ and the group inverse A$^{\#}$ by matrix operations rather than by row operations.

Under a perturbation $\boldsymbol{E}$ when $\boldsymbol{\pi}^T$ leads to $\bar{\boldsymbol{\pi}}^T = (\boldsymbol{\pi}^T(I - \boldsymbol{E}A^{\#})^{-1})$, if $\Pi = \boldsymbol{e}\boldsymbol{\pi}^T$ and $\overline{\Pi} = \boldsymbol{e}\overline{\boldsymbol{\pi}}^T$ then $\overline{\Pi} = \Pi(I - \boldsymbol{E}A^{\#})^{-1}$.

Now under the perturbation $\boldsymbol{E} = \boldsymbol{e}_i\boldsymbol{b}_i^T$ to the *i*-th row with $\boldsymbol{b}_i^T\boldsymbol{e} = 0$, yields,

$$\overline{\Pi} = \Pi\left[I + \frac{1}{1 - \boldsymbol{b}_i^T A^{\#}\mathbf{e}_i}\boldsymbol{e}_i\boldsymbol{b}^T A^{\#}\right] \text{ and } \overline{A}^{\#} = (I - \overline{\Pi})A^{\#}\left(I + \frac{1}{1 - \boldsymbol{b}_i^T A^{\#}\boldsymbol{e}_i}\boldsymbol{e}_i\boldsymbol{b}_i^T A^{\#}\right).$$

This leads to the following procedure. (For more details see Algorithm 3 in [18]).

Procedure 5: (Group inverse by matrix updating – Pert AL3)

(i) Let $P_0 = \boldsymbol{e}\boldsymbol{e}^T/m$, implying $\Pi_0 = \boldsymbol{e}\boldsymbol{e}^T/m$, $A_0^{\#} = I - \boldsymbol{e}\boldsymbol{e}^T/m$.

(ii) For $i = 1, 2, ..., m$, let $\boldsymbol{p}_i^T = \boldsymbol{e}_i^T P$, $\boldsymbol{b}_i^T = \boldsymbol{p}_i^T - \boldsymbol{e}^T/m$,

$$S_i = I + \frac{1}{1 - \boldsymbol{b}_i^T A_{i-1}^{\#}\boldsymbol{e}_i}\boldsymbol{e}_i\boldsymbol{b}_i^T A_{i-1}^{\#}, \quad \Pi_i = \Pi_{i-1}S_i, \quad \mathrm{A}_i^{\#} = (I - \Pi_i)A^{\#}{}_{i-1}S_i.$$

(iii) At $i = m$, let S = S_m then $\Pi = \Pi_{m-1}S$, $A^{\#} = (I - \Pi)A_{m-1}^{\#}S$.

(iv) Compute $M = [I - A^{\#} + EA_d^{\#}]D$, where $E = [1]$ and $D = (\Pi_d)^{-1}$.

We now give three interrelated procedures, each with different starting conditions, based on updating simple generalised inverses of $I - P_0$ that lead to simple computations for the stationary probabilities, the group inverse and the mean first passage time matrix.

From Theorem 2.3(c)), if we choose a g-inverse G of $I - P$ with the property that $G\boldsymbol{e} = g\boldsymbol{e}$, by taking G of the form $G = [I - P + \boldsymbol{e}\boldsymbol{\beta}^T]^{-1}$, then we have expressions for $\boldsymbol{\pi}^T$ that are given as $\boldsymbol{\pi}^T = \boldsymbol{\beta}^T[I - P + \boldsymbol{e}\boldsymbol{\beta}^T]^{-1}$. Further we have a simple form of the *MFPT* matrix M given by eqn. (2.3). While it is easy to find an expression for the group inverse of $I - P$ as $A^{\#} = (I - \boldsymbol{e}\boldsymbol{\pi}^T)G$ we don't actually require that step to find expressions for M.

In Hunter [15] we explored the properties of some generalized inverses of this form. For the three procedures to follow we use, successively, the special forms,

$G_e \equiv \left[I - P + \frac{\boldsymbol{e}\boldsymbol{e}^T}{m} \right]^{-1}$, $G_{e1} \equiv [I - P + \boldsymbol{e}\boldsymbol{e}_1^T]^{-1}$ and $G_{ee} \equiv [I - P + \boldsymbol{e}\boldsymbol{e}^T]^{-1}$, and the Sherman-Morrison matrix inversion formula ([22]). The starting conditions for each procedure are different and, although we carry out similar recursions, we have different expressions for the stationary probability vector $\boldsymbol{\pi}^T$ but identical calculation procedures for the *MFPT* matrices. Once again the reader is referred to [18] for full details.

This leads to three further algorithms – Algorithms 4A, 4B, and 4C in [18]. They are all variants of the generic recursion given by (*ii*) with identical steps (*iv*) and (*v*) as in Procedure 6.

Procedure 6: (Update using G_e - Pert AL4A)

(*i*) Start with $K_0 = I$.

(*ii*) For $i = 1, 2, ..., m$, let $\boldsymbol{p}_i^T = \boldsymbol{e}_i^T P$, $\boldsymbol{b}_i^T = \boldsymbol{p}_i^T - \boldsymbol{e}^T / m$,

$$K_i = K_{i-1}\left(I + C_i\right), where\ k_i = 1 - \boldsymbol{b}_i^T K_{i-1} \boldsymbol{e}_i\ and\ C_i = \frac{1}{k_i} \boldsymbol{e}_i \boldsymbol{b}_i^T K_{i-1}.$$

(*iii*) At $i = m$, let $K = K_m$ and then compute $\boldsymbol{\pi}^T = \frac{1}{m} \boldsymbol{e}^T K$

(*iv*) Compute $M = [I - K + EK_d]D$, where $E = [1]$ and $D = (\Pi_d)^{-1}$.

Procedure 7: **(Update using G_{e1} - Pert AL4B)**

(*i*) Start with $K_0 = I + \boldsymbol{e}\left(\frac{\boldsymbol{e}^T}{m} - \boldsymbol{e}_1^T \right)$.

(ii) Carry out Step *(ii)* of Procedure 6.

(*iii*) At $i = m$, let $K = K_m$ and then compute $\boldsymbol{\pi}^T = \boldsymbol{e}_1^T K$.

(iv) Carry out Step *(iv)* of Procedure 6, to compute *M*.

Procedure 8: **(Update using G_{ee} – Pert AL4C)**

(*i*) Start with $K_0 = I - \left(\frac{m-1}{m^2} \right) \boldsymbol{e}\boldsymbol{e}^T$.

(ii) Carry out Step (*ii*) of Procedure 6.

(*iii*) At $i = m$, let $K = K_m$ and then compute $\boldsymbol{\pi}^T = \boldsymbol{e}^T K$.

(iv) Carry out step *(iv)* of Procedure 6, to compute *M*.

Section 4: Computation of *MFPTs* using Hunter Extended *GTH* (*EGTH*) procedure

The details of this *EGTH* procedure are given in Hunter [17] to which the reader is referred. We make use of the *GTH* procedure of Grassman, Taksar and Heyman ([3]) (or the equivalent state reduction procedure by Sheskin ([23]) for finding the stationary probability vector $\boldsymbol{\pi}$. Let us provide some details that serves to introduce some additional notation.

We start with the given transition matrix P (= $P^{(m)}$) of the irreducible *MC* $\{X_k^{(m)}, k \geq 0\}$ with state space $S = \{1,2,\ldots, m\} = S_m$. The general idea is to reduce the state space, one state at a time successively removing states $m - 1$, m- 2, … until we are left with a single state 1. Once state 1 is reached the state space is expanded one state at a time i.e. insert states 2, …, to finally insert state m.

Suppose we reach the stage where we have n states $S_n = \{1, 2, \ldots n\}$ with *MC* $\{X_k^{(n)}, k \geq 0\}$ and transition matrix $P^{(n)}$, then it is easily shown during the state reduction process that the elements of $P^{(n-1)} = \left[p_{ij}^{(n-1)} \right]$ are related as

$$p_{ij}^{(n-1)} = p_{ij}^{(n)} + \frac{p_{in}^{(n)} p_{nj}^{(n)}}{S(n)}, \quad 1 \leq i \leq n-1, \; 1 \leq j \leq n-1, \tag{4.1}$$

where $S(n) = 1 - p_{nn}^{(n)} = \sum_{j-1}^{n-1} p_{nj}^{(n)}$. Note that the transition probabilities of the reduced *MC* can all be obtained without carrying out any subtraction. The *MC* $\{X_k^{(n-1)}, k \geq 0\}$ on the reduced state space, S_{n-1} is the "censored" *MC* (see [2]), i.e. the *MC* restricted to the states of S_{n-1}. Further the irreducibility of the reduced state *MC* is retained. One can derive relationships between the stationary distributions of the respective *MCs*, $\{\pi_i^{(n)}, i \in S_n\}$ for $\{X_k^{(N)}, k \geq 0\}$. In particular, it can be shown

$$\pi_i^{(n-1)} = \frac{\pi_i^{(n)}}{1 - \pi_n^{(n)}} = \frac{\pi_i^{(n)}}{\sum_{k=1}^{n-1} \pi_k^{(n)}}, \quad 1 \leq i \leq n-1.$$

Similarly when we expand the state space we can show that

$$\boldsymbol{\pi}^{(n)} \equiv \left(\pi_1^{(n)}, \ldots, \pi_n^{(n)} \right) = c_{n-1} \left(\pi_1^{(n-1)}, \ldots, \pi_{n-1}^{(n-1)}, \frac{\sum_{i=1}^{n-1} \pi_i^{(n-1)} p_{in}^{(n)}}{S(n)} \right),$$

where c_{n-1} is determined from the fact that $\sum_{i=1}^{n} \pi_i^{(n)} = 1$.
From these results we have the following algorithm.

***GTH* Procedure for computing the stationary probabilities of a *MC*:**
Let *MC* $\{X_k^{(m)}, k \geq 0\}$ be finite irreducible *MC* with state space $S_m = \{1, 2, \ldots, m\}$ and transition matrix $P = P^{(m)} = \left[p_{ij}^{(m)} \right]$. Let $\{\pi_i^{(m)}\}$ be its stationary probabilities.
Step 1. Compute, successively for $n = m, m - 1, \ldots, , 3$,

$p_{ij}^{(n-1)} = p_{ij}^{(n)} + \frac{p_{in}^{(n)} p_{nj}^{(n)}}{S(n)}, \quad 1 \leq i \leq n-1, \; 1 \leq j \leq n-1$ where $S(n) = \sum_{j=1}^{n-1} p_{nj}^{(n)}$.

Step 2. Set r_1=1 and compute successively for $n = 2, \ldots, m$, $r_n = \sum_{i=1}^{n-1} r_i p_{in}^{(n)} \Big/ S(n)$.

Step 3. Compute, for $i = 1, 2, \ldots, m$, $\pi_i^{(m)} = r_i \Big/ \sum_{j=1}^{m} r_j$. □

In extending this algorithm to find the *MFPTs,* Kohlas showed ([20]) that it is more natural to consider the process as a Markov renewal process (*MRP*), $\{(X_k^{(n)},T_k^{(n)}),k\geq 0\}$, where $\{X_k^{(n)},k\geq 0\}$ is the embedded *MC* when the state space is $S_n=\{1,..,n\}$ and $T_k^{(n)}$ is time that the process stays in the state before making the next transition. Let $\mu_i^{(n)}=E[T_k^{(n+1)}-T_k^{(n)}\mid X_k^{(n)}=i]$ be the expected holding time in state *i* when the state space is S_n. When the process is censored by eliminating state *n* the mean holding time vector eliminates that state and reduces to a smaller (*n*-1)-dimension vector as

$$\boldsymbol{\mu}^{(n-1)T}=(\mu_1^{(n-1)},...,\mu_{n-1}^{(n-1)}) \text{ where } \mu_i^{(n-1)}=\mu_i^{(n)}+\frac{p_{in}^{(n)}\mu_n^{(n)}}{S(n)},\ 1\leq i\leq n-1.$$

Under the *MC* setting, which we assume in this paper, initially $\mu_i^{(m)}=1$ for all $i\in S_m$. In [17] it is shown how this influences the *MFPTs* showing, in particular, that

$$m_{ij}=\frac{\mu_i^{(i)}+\sum_{k=1,k\neq j}^{i-1}p_{ik}^{(i)}m_{kj}}{S(i)},\ 3\leq i\leq m,1\leq j\leq i-1,\ \text{with } m_{21}=\frac{\mu_2^{(2)}}{S(2)},\text{and}$$

$$m_{ii}=\mu_i^{(i)}+\sum_{k=1,}^{i-1}p_{ik}^{(i)}m_{ki},\ 2\leq i\leq m,\ \text{with } m_{11}=\mu_1^{(1)}.$$

The expressions for m_{ij}, for $1\leq i\leq m-1,\ i+1\leq j\leq m$ are much more complicated; (see [17]). However, by focussing primarily on the terms m_{i1} for $1\leq i\leq m$, i.e. the first column of the matrix of *MFPTs,* we can produce a simple algorithmic procedure.

Procedure 9: (*EGTH* – Hunter Extended *GTH* Procedure)

Let $\{X_k^{(m)},k\geq 0\}$ be a finite irreducible *MC* with state space $S_m=\{1,2,\ldots,m\}$ and transition matrix $P\equiv P^{(m)}\equiv\left[p_{ij}^{(m)}\right]$.

Step 1(i): Carry out step 1 of the *GTH* Procedure, i.e.

Compute, successively for $n=m, m-1,\ldots,3$,

$$p_{ij}^{(n-1)}=p_{ij}^{(n)}+\frac{p_{in}^{(n)}p_{nj}^{(n)}}{S(n)},\ 1\leq i\leq n-1,\ 1\leq j\leq n-1 \text{ where } S(n)=\sum_{j=1}^{n-1}p_{nj}^{(n)}.$$

Step 1(*ii*): Compute, successively for $n=m, m-1,\ldots,,3,2$,

$$\mu_i^{(n-1)}=\mu_i^{(n)}+\frac{\mu_n^{(n)}p_{in}^{(n)}}{S(n)},\ 1\leq i\leq n-1,\text{where } \boldsymbol{\mu}^{(m)T}=(\mu_1^{(m)},....,\mu_m^{(m)})=(1,...,1).$$

Step 1(*iii*): Compute the $m\times 1$ column vector $\boldsymbol{m}_m^{(1)(m)}=(m_{i1})$,

where $m_{11}=\mu_1^{(1)}$, $m_{21}=\frac{\mu_2^{(2)}}{S(2)}$, and for $i=3,,,,,m,\ m_{i1}=\frac{\mu_i^{(i)}+\sum_{k=2}^{i-1}p_{ik}^{(i)}m_{k1}}{S(i)}$.

Thus, starting with $P^{(m)}\equiv P^{(m)(1)}$, we can obtain the entries of the first column of $M=[m_{ij}]=\left(m_m^{(1)(m)},m_m^{(2)(m)},...,m_m^{(m)(m)}\right)$.

The procedure that follows to find the other *MFPTs* is to permute the state space to $S_m^{(2)}=\{2,3,\ldots,m,1\}$ and do this successively finishing up with $S_m^{(m)}=\{m,1,2,\ldots m-1\}$. This can be effected by permuting the elements of the transition matrix. For example, for $S_m^{(2)}$ we can do this by moving the elements of first column of $P^{(m)}$ to after the *m*-th

column, followed by moving the first row to the last row, to obtain a new transition matrix $P^{(m)(2)}$. One of the easier ways to do this in *MatLab* is to note that
$P^{(m)(2)}\,(mod(row + m - 2, m) + 1, mod(col + m - 2, m) + 1) = P^{(m)(1)}\,(row, col)$.

Step 2: For $k = 2, 3, 4, \ldots, m-1, m$.
(*i*) Repeat Step 1(*i*) with $P^{(m)} = P^{(m)(k)}$.
(*ii*) Repeat Step 1(*ii*) with $\boldsymbol{\mu}^{(k)(m)} = \boldsymbol{\mu}^{(m)} = (1,,1,..,1)$.
(*iii*) Repeat Step 1(*iii*) to calculate the $m \times 1$ column vector $\bar{\boldsymbol{m}}_m^{(k)(m)T} =$
$\left(m_{kk}, m_{k+1,k}, ..m_{m,k}, m_{1,k}, ..., m_{k+1,k}\right)$.

Step 3: Combine the results of the Steps 1(*iii*) and 2(*iii*) to find *M* as follows.
Let $\bar{M} = (m_m^{(1)(m)}, \bar{m}_m^{(2)(m)}, ..., \bar{m}_m^{(m)(m)})$ and reorder the elements of $\bar{M}$ to obtain
$M = (m_m^{(1)(m)}, m_m^{(2)(m)}, ..., m_m^{(m)(m)})$. This can be carried out in *MatLab* by noting that for each row and column entry, $\bar{M}(\text{mod}(row + col - 2, m) + 1, col) = M(row, col)$.

A key observation is that the *EGTH* algorithm retains calculation accuracy as no subtractions are involved.

Further, the stationary probabilities do not need to be computed in advance and can be found directly as inverses of the mean recurrence times. Once again, no subtraction operation need be performed.

Section 5: Computation of *MFPTs* using the Heyman *FUND* algorithm and its modifications

In carrying out Step 1 of the *EGTH* algorithm observe that the elements for $p_{ij}^{(n-1)}$ in the block upper left hand $(n-1)\times(n-1)$ corner of the transition matrix are based only on the elements $p_{ij}^{(n)}, p_{in}^{(n)}, p_{nj}^{(n)}$, $1 \le i \le n-1$, $1 \le j \le n-1$. This means that we can in effect overwrite the elements of the transition matrix that are not required in the future. At the conclusion of the reduction process we are this left with a matrix of elements $\bar{P} = \left[\,\bar{p}_{ij}\,\right]$,

where $\overline{p_{ij}} = \begin{cases} p_{ij}^{(j)} = u_{ij}, & 1 \le i < j \le m, \\ p_{ii}^{(i)} = d_{ii}, & 1 \le i = j \le m, \\ p_{ij}^{(i)} = l_{ij}, & 1 \le j < i \le m; \end{cases}$

so that $\bar{P} = \bar{U} + \bar{D} + \bar{L}$ where $\bar{U}$ is strictly upper triangular, $\bar{L}$ is strictly lower triangular and $\bar{D}$ is a diagonal matrix.

Observe from (4.1) that $p_{ij}^{(n-1)} = p_{ij}^{(n)} + \dfrac{p_{in}^{(n)} p_{nj}^{(n)}}{S(n)}$, $1 \le i \le n-1,\ 1 \le j \le n-1,$ so that

$$p_{ij}^{(n-2)} = p_{ij}^{(n-1)} + \frac{p_{i,n-1}^{(n-1)} p_{n-1,j}^{(n-1)}}{S(n-1)} = p_{ij}^{(n)} + \frac{p_{in}^{(n)} p_{nj}^{(n)}}{S(n)} + \frac{p_{i,n-1}^{(n-1)} p_{n-1,j}^{(n-1)}}{S(n-1)},\ 1 \le i \le n-2,\ 1 \le j \le n-2.$$

It is easy to establish, by considering $t = n - 3, \ldots, t = n - k$, that

$$p_{ij}^{(t)} = p_{ij}^{(n)} + \sum_{k=t+1}^{n} \frac{p_{ik}^{(k)} p_{kj}^{(k)}}{S(k)},\ 1 \le i \le t \le n-1,\ 1 \le j \le t \le n-1.$$

Since $p_{ij}^{(m)} = p_{ij}$ it is easy to see that if, for $k = 2, \ldots, m$, we define $q_{kk} = 1/S(k)$,

$p_{ij}^{(t)} = p_{ij} + \sum_{k=t+1}^{m} u_{ik} q_{kk} l_{kj}$ for $1 \le i \le t \le m-1,\ 1 \le j \le t \le m-1$ with $p_{ij}^{(t)} = p_{ij}$ for $i = m$ *or* $j = m$.

Thus $\overline{P} = P + \overline{U}\overline{Q}\overline{L}$ where $\overline{Q} \equiv diag(q_{11}, q_{22}, \ldots, q_{mm})$. Note that at this stage q_{11} can be arbitrarily defined. The first column and last row of $\overline{U}$ are empty and the first row and column of $\overline{L}$ are also empty. Further, since $S(k) = \sum_{j=1}^{k-1} p_{kj}^{(k)} = 1 - p_{kk}^{(k)}$,

$\overline{D} = diag(p_{11}^{(1)}, p_{22}^{(2)}, \ldots, p_{mm}^{(m)}) = diag(1, 1-S(2), \ldots, 1-S(m))$ implying $I - \overline{D} = diag(0, S(2), \ldots, S(m))$.
Let $\overline{S} = diag(1, S(2), \ldots, S(m)) = E_{11} + I - \overline{D}$ so that $\overline{D} = E_{11} + I - \overline{S}$. We define $q_{11} = 1$ so that $\overline{S}^{-1} = \overline{Q}$. From these results we establish the following theorem.

Theorem 5.1: For an irreducible transition matrix P, the Markovian kernel $I - P$ can be factored into a UL form where L is a lower triangular matrix and U is an upper triangular matrix, i.e. $I - P = UL$.
In particular, if $\overline{P} = \overline{U} + \overline{D} + \overline{L}$ is the matrix of overwritten elements of P from the *GTH* algorithm, $U = \overline{U}\overline{S}^{-1} - I$ and $L = \overline{L} - (I - \overline{D})$ where $\overline{S} = E_{11} + (I - \overline{D})$.

Proof: From the results above

$$I - P = I - \overline{P} + \overline{U}\overline{Q}\overline{L} = I - \overline{U} - \overline{D} - \overline{L} + \overline{U}\overline{S}^{-1}\overline{L} = I - \overline{D} - \overline{U} + (\overline{U}\overline{S}^{-1} - I)\overline{L}$$

i.e. $I - P = \overline{S} - E_{11} - \overline{U} + (\overline{U}\overline{S}^{-1} - I)\overline{L}$ since $\overline{S} - E_{11} = I - \overline{D}$.

Now $(\overline{U}\overline{S}^{-1} - I)(E_{11} - \overline{S}) = \overline{U}\overline{S}^{-1}E_{11} - \overline{U} - E_{11} + \overline{S} = \overline{S} - E_{11} - \overline{U}$

since $(\overline{U}\overline{S}^{-1}E_{11} = \overline{U}diag(q_{11}, q_{22}, \ldots, q_{mm})E_{11} = \overline{U}E_{11} = 0$ (since $u_{11} = 0$).

Thus $I - P = \left(\overline{U}\overline{S}^{-1} - I\right)\left(\overline{L} - \overline{S} + E_{11}\right) = UL$, where

$U \equiv \overline{U}\overline{S}^{-1} - I$ is upper triangular and $L \equiv \overline{L} - \overline{S} + E_{11} = \overline{L} - (I - \overline{D})$ is lower triangular.

□

Grassman [4] first explored an UL factorisation of $I - P$ based upon the *GTH* algorithm. A version of this UL factorisation was used by Heyman [7] to produce his *FUND* algorithm to compute Z, the fundamental matrix of irreducible Markov chains. The proof

given above is modified, due to some arbitrariness in the choice of the $\overline{Q}$ matrix, through a possible choice of q_{11}.

Our choice for $\overline{Q}$ and hence for $\overline{S}$ leads to U having all the elements of its diagonal as -1 and the other elements strictly upper triangular. This leads to U having determinant $(-1)^m$ and consequently implying the non-singularity of U. L has all the elements of its first row as 0.

Heyman [7] uses the UL factorisation to find an expression for Z. For completeness we incorporate the results of his Theorem 1within our Theorem 5.2 below but extend his results and give a non-constructive proof.

Theorem 5.2: Let P be the transition matrix of an irreducible finite MC, $\boldsymbol{\pi}^T$ its stationary probability vector and $\Pi = \boldsymbol{e}\boldsymbol{\pi}^T$.
(i) If X is any solution of

$$(I - P)X = I - \Pi, \tag{5.1}$$

then X is a one-condition generalised inverse of $I - P$ and satisfies the property that

$$X\boldsymbol{e} = x\boldsymbol{e}, \text{ where } x \text{ is a constant.} \tag{5.2}$$

(*ii*) If X is a solution of (5.1) then $A^{\#}$, the group inverse of $I - P$, is given by

$$A^{\#} = (I - \Pi)X, \tag{5.3}$$

and Z, the fundamental inverse of $I - P$, is given by

$$Z = \Pi + (I - \Pi)X. \tag{5.4}$$

Proof:
(i) Observe that from (5.1) and (1.2), $(I - P)X(I - P) = (I - \boldsymbol{e}\boldsymbol{\pi}^T)(I - P) = I - P$, implying that X is a one-condition generalised inverse of $I - P$. Further, from (5.1) and (1.2), $(I - P)X\boldsymbol{e} = \boldsymbol{e} - \boldsymbol{e}\boldsymbol{\pi}^T\boldsymbol{e} = \boldsymbol{0}$, implying that $X\boldsymbol{e}$ is a right eigenvector of $I - P$ and hence must be a multiple of $\boldsymbol{e}$.
(ii) From Theorem 6.3 of [11] or Corollary 4.6 of [13], if G is any g-inverse of $I - P$, when P is irreducible, then $(I - \Pi)G(I - \Pi) = A^{\#}$. Taking $G = X$ observe that $(I - \Pi)X(I - \Pi) = (I - \Pi)X - (I - \Pi)X\boldsymbol{e}\boldsymbol{\pi}^T = (I - \Pi)X - x\boldsymbol{e}\boldsymbol{\pi}^T + x\boldsymbol{e}\boldsymbol{\pi}^T\boldsymbol{e}\boldsymbol{\pi}^T = (I - \Pi)X$ leading to (5.3). Z, the fundamental inverse of $I - P$, is given by $Z = [I - P + \Pi]^{-1}$
$= A^{\#} + \Pi = \Pi + (I - \Pi)X$, leading to (5.4).

□

With $I - P = UL$, equation (5.1) can be solved in steps.
First let

$$LX = Y, \tag{5.5}$$

implying that

$$UY = I - \Pi. \tag{5.6}$$

We first solve, from equation (5.6), Y, uniquely, by backward substitution. In Matlab we use the procedure $Y = U\backslash(I - \Pi)$. Note that for all $j = 1, \ldots, m$, $\boldsymbol{y}_j^{(r)T}\boldsymbol{e} = \sum_{i=1}^{m} y_{ij} = 0$.
Further, from equation (5.5), since $\boldsymbol{e}_1^T L = \boldsymbol{0}^T$, we have that $\boldsymbol{e}_1^T Y = \boldsymbol{0}^T$, and we conclude that the first row of Y, $\boldsymbol{y}_1^{(r)T} = (y_{11}, \ldots, y_{1m})$, consists of zero elements.

Since we may take any one-condition g-inverse of $I - P$, we may take the first row of X as the zero vector. Thus we may partition L, X and Y in block form as

$$LX = \begin{bmatrix} 0 & \boldsymbol{0}^T \\ \boldsymbol{l}_1^{(c)} & L_1 \end{bmatrix} \begin{bmatrix} 0 & \boldsymbol{0}^T \\ \boldsymbol{x}_1^{(c)} & X_1 \end{bmatrix} = \begin{bmatrix} 0 & \boldsymbol{0}^T \\ \boldsymbol{y}_1^{(c)} & Y_1 \end{bmatrix} = Y,$$

implying that $L_1\boldsymbol{x}_1^{(c)} = \boldsymbol{y}_1^{(c)}$ and $L_1X_1 = Y_1$, or equivalently $L_1(\boldsymbol{x}_1^{(c)}, X_1) = (\boldsymbol{y}_1^{(c)}, Y_1)$.
Thus if $\hat{X} = (\boldsymbol{x}_1^{(c)}, X_1)$, $\hat{Y} = (\boldsymbol{y}_1^{(c)}, Y_1)$, then we need to solve $L_1\hat{X} = \hat{Y}$.

Procedure 10: (Heyman *FUND* Algorithm for *M* using *Z*).

1. Start with P and use the *GTH* algorithm to compute $\boldsymbol{\pi}^T$.
2. Use the decomposition of Theorem 5.1 finding $\overline{P}$ and hence U and L.
3. Solve $UY = \mathrm{I} - \Pi$, where $\Pi = \boldsymbol{e}\boldsymbol{\pi}^T$, by back substitution.
4. Solve $L_1\hat{X} = \hat{Y}$, by forward substitution.
5. Let $X = \begin{bmatrix} \boldsymbol{0}^T \\ \hat{X} \end{bmatrix}$.
6. Compute $Z = \Pi + (I - \Pi)\mathrm{X}$.
7. Compute $M = [\, I - Z + EZ_d]D$ where $D = \left(\Pi_d\right)^{-1}$.

Heyman's *FUND* algorithm for finding the *MFPT*'s can be modified by noting that instead of computing Z one can compute the group inverse, reducing the number of calculations required in Step 6 of Procedure 10 as follows.

Procedure 11: (Modified Heyman *FUND* Algorithm for *M* using $A^{\#}$).

1. Carry out steps 1 to 5 of Procedure 10.
2. Compute $A^{\#} = (I - \Pi)X$.
3. Compute $M = [\, I - A^{\#} + EA_d^{\#}\,]D$ where $D = \left(\Pi_d\right)^{-1}$.

Actually one doesn't need to compute either Z or $A^{\#}$ since X is a one-condition inverse with the property that $X\boldsymbol{e} = \boldsymbol{0}$, ($x = 0$ in Theorem 5.2 (*i*), since X is chosen to have the first row the zero vector) and thus from Theorem 2.3 (*c*) the following procedure is justified. Note that Heyman also observed this computational benefit for finding M in the final section of his paper, [7].

Procedure 12: (Modified Heyman *FUND* Algorithm for *M* using *X*).

1. Carry out steps 1 to 5 of Procedure 10.
2. Compute $M = [\, I - X + EX_d\,]D$ where $D = \left(\Pi_d\right)^{-1}$.

Section 6: Computational comparisons

For our numerical computations and comparisons we coded each algorithm using *MatLab* (64-bit version R2015b on a *MacBook Air* computer) and used various test problems, which have previously been considered in the literature. *MatLab* was run in both single and double precision to enable us to compute and compare the matrices $M(S) = [m_{ij}(S)]$ and $M(D) = [m_{ij}(D)]$ for each procedure and test problem.

We use the following test problems which were introduced by Harrod & Plemmons ([5]) They were initially introduced as poorly conditioned examples for computing the stationary distribution of the underlying irreducible *MC* but have been used as examples for testing various different algorithms for computing *M*, the matrix of *MFPTs*, ([8], [9]). While the dimensions of the state space are relatively small, the test problems lead to some computational difficulties.

TP1: (As modified by Heyman and Reeves ([9]). The original version of TP1, given in [5] related to a 10-state MC however it was shown, by Heyman [6], that four of the states were in fact transient and the irreducible sub chain was identified as

$$\begin{bmatrix} .1 & .6 & 0 & .3 & 0 & 0 \\ .5 & .5 & 0 & 0 & 0 & 0 \\ .5 & .2 & 0 & 0 & .3 & 0 \\ 0 & .7 & 0 & .2 & 0 & .1 \\ .1 & 0 & .8 & 0 & 0 & .1 \\ .4 & 0 & .4 & 0 & 0 & .2 \end{bmatrix}$$

TP2: A typo for the original problem for element (1, 5) was identified and corrected in [9]. The test problem is also known as the 8 X 8 Courtois matrix and was also considered in a paper by Benzi, [1].

$$\begin{bmatrix} .85 & 0 & .149 & .0009 & 0 & .00005 & 0 & .00005 \\ .1 & .65 & .249 & 0 & .0009 & .00005 & 0 & .00005 \\ .1 & .8 & .09996 & .0003 & 0 & 0 & .0001 & 0 \\ 0 & .0004 & 0 & .7 & .2995 & 0 & .0001 & 0 \\ .0005 & 0 & .0004 & .399 & .6 & .0001 & 0 & 0 \\ 0 & .00005 & 0 & 0 & .00005 & .6 & .2499 & .15 \\ .00003 & 0 & .00003 & .00004 & 0 & .1 & .8 & .0999 \\ 0 & .00005 & 0 & 0 & .00005 & .1999 & .25 & .55 \end{bmatrix}.$$

TP3:

$$\begin{bmatrix} 0.999999 & 1.0\ E-07 & 2.0\ E-07 & 3.0\ E-07 & 4.0\ E-07 \\ 0.4 & 0.3 & 0 & 0 & 0.3 \\ 5.0\ E-07 & 0 & 0.999999 & 0 & 5.0\ E-07 \\ 5.0\ E-07 & 0 & 0 & 0.999999 & 5.0\ E-07 \\ 2.0\ E-07 & 3.0\ E-07 & 1.0\ E-07 & 4.0\ E-07 & 0.999999 \end{bmatrix}.$$

TP4 and variants:

$\mathbf{TP41}: \varepsilon = 1.0E-01;\ \mathbf{TP42}: \varepsilon = 1.0E-03; \mathbf{TP43}: \varepsilon = 1.0E-05;\ \mathbf{TP44}: \varepsilon = 1.0E-07$

$$\begin{bmatrix} .1-\varepsilon & .3 & .1 & .2 & .3 & \varepsilon & 0 & 0 & 0 & 0 \\ .2 & .1 & .1 & .2 & .4 & 0 & 0 & 0 & 0 & 0 \\ .1 & .2 & .2 & .4 & .1 & 0 & 0 & 0 & 0 & 0 \\ .4 & .2 & .1 & .2 & .1 & 0 & 0 & 0 & 0 & 0 \\ .6 & .3 & 0 & 0 & .1 & 0 & 0 & 0 & 0 & 0 \\ \varepsilon & 0 & 0 & 0 & 0 & .1-\varepsilon & .2 & .2 & .4 & .1 \\ 0 & 0 & 0 & 0 & 0 & .2 & .2 & .1 & .3 & .2 \\ 0 & 0 & 0 & 0 & 0 & .1 & .5 & 0 & .2 & .2 \\ 0 & 0 & 0 & 0 & 0 & .5 & .2 & .1 & 0 & .2 \\ 0 & 0 & 0 & 0 & 0 & .1 & .2 & .2 & .3 & .2 \end{bmatrix}$$

We compute, with the specified transition matrix for each test problem, the following errors for the *MFPT* matrix, $M = \left[m_{ij} \right]$, under both double and single precision:

Minimum absolute residual error, MINRE $= \min_{1 \le i \le m, 1 \le j \le m} \left| m_{ij} - \sum_{k \ne j} p_{ik} m_{kj} - 1 \right|$,

Maximum absolute residual error, MAXRE $= \max_{1 \le i \le m, 1 \le j \le m} \left| m_{ij} - \sum_{k \ne j} p_{ik} m_{kj} - 1 \right|$,

and the *overall residual error, ORE* $= \sum_{i=1}^{m} \sum_{j=1}^{m} \left| m_{ij} - \sum_{k \ne j} p_{ik} m_{kj} - 1 \right|$.

The accuracy of each algorithm was also evaluated in terms of the *minimum error, the maximum error and the relative errors between the double and single precision computations as*

MINE(S, D) $= \min_{1 \le i \le m,\ 1 \le j \le m} \left| m_{ij}(S) - m_{ij}(D) \right|$,

MAXE(S, D) $= \max_{1 \le i \le m,\ 1 \le j \le m} \left| m_{ij}(S) - m_{ij}(D) \right|$,

and *REL(S, D)* $= \sum_{i=1}^{m} \sum_{j=1}^{m} \left| m_{ij}(S) - m_{ij}(D) \right|$.

Following Heyman and Reeves ([9]) and Heyman and O'Leary ([8]), if one regards the double precision result as the "true" result and the single precision result as the "computed" result, then the *number of (extra) accurate digits* can be defined as the

overall average of $-\log_{10}\left|\frac{result_{true}-result_{computed}}{result_{true}}\right|$. We compute this statistic for each of our procedures and test problems.

We identify the following procedures in the tables and charts to follow.

Procedure 1: (Standard method)
Procedure 2: (Simple method)
Procedure 3: (G-inverse update – Pert AL1)
Procedure 4: (Group inverse update – Pert AL2)
Procedure 5: (Group inverse by matrix updating – Pert AL3)
Procedure 6: (Update using G_e – Pert AL4A)
Procedure 7: (Update using G_{e1} – Pert AL4B)
Procedure 8: (Update using G_{ee} – Pert AL4C)
Procedure 9: (*EGTH* – Hunter Extended *GTH* Procedure)
Procedure 10: (Heyman *FUND* Algorithm for *M* using *Z*).
Procedure 11: (Modified Heyman *FUND* Algorithm for *M* using $A^{\#}$).
Procedure 12: (Modified Heyman *FUND* Algorithm for *M* using *X*).

Appendix 1 gives a table of all the *MatLab* results for the error calculations for all test problems and procedures.

An arXiv.com version of this paper gives an additional appendix with a set of Excel charts for the errors for all procedures and test problems.

We make some general comments regarding the computations and the results.

Firstly, if the computation of the minimum absolute residual error (*MINRE*) yields a zero (either under single or double precision) indicates that at least one *MFPT* is computed exactly. Under single precision only Procedures 2, 9 and 12 yield *MINRE* zero for all test problems whilst under double precision for all procedures and test problems this error is always zero. Thus single precision is not generally recommended as a suitable computation procedure unless one is using the restricted set of Procedures 2, 9, and 12.

For those procedures with minimal *MAXRE* errors all absolute residual errors are kept under control. No one single procedure does that to achieve the smallest *MAXRE* for any given test problem. Procedures 1, 3, 5, 6, 8, 9 under single precision and Procedures 1, 2, 3, 6, 7, 9, 10, 11, 12 under double precision each generate a minimum value for at least one test problem.

The overall residual error, *ORE*, is perhaps a better indicator of accuracy as a small *ORE* indicates that the sum of all the absolute residual errors are minimized. Minimum values of *ORE* are achieved by the following procedures for specific test problems.
Under single precision: Procedure 12 (*TP*1), Procedure 9 (*TP*2, *TP*3, *TP*41, *TP*42, *TP*43), Procedure 3 (*TP*44).

Under double precision: Procedure 11 (*TP*1), Procedure 9 (*TP*2, *TP*3, *TP*41,*TP*42, *TP*43), Procedure 3 (*TP* 44).
This does suggest that Procedure 9 (Hunter's Extended *GTH* Procedure) appears to generally give the most accurate results. When the other procedures give a smaller error, Procedure 9 is the second most accurate procedure (except for *TP*1 under single precision when it doesn't give a very accurate result at all). Chart 6.1 gives the overall residual errors, as given in Table A.6 for all procedures and test problems, under double precision.

Chart 6.1: Overall Residual Errors, under double precision, all procedures and test problems

See Chart B.6: ORE under Double Precision

Heyman and Reeves ([9]) and Heyman and O'Leary ([8]) computed the accurate digits statistic for the above set of test problems when computing the *MFPT* matrix. In both of these papers their results were displayed in figures and no actual numerical results were tabulated. We computed this statistic for each the seven test problems and the twelve procedures, achieving the results given in Appendix 1: Table A.10

We comment on *TP*1. This test problem has some unique features in that it is possible to deduce exact results for three *MFPT's*. In particular, it can be shown that $m_{21} = 2$, $m_{43} = 160.5$ and $m_{53} = 26.3$. If one computes the *MFPT* matrices we find that, under double precision, all the twelve procedures obtain these exact three results. However, under single precision, only Procedure 1 and Procedure 12 yield all three exact results, while Procedure 2 gives the exact results for m_{21} and m_{43}, and Procedures 9, 10 and 11 yield the exact result for m_{21}. Thus when calculating the average number of accurate digits we must omit the results when the *MFPTs* under single and double precision are the same, as the logarithm of zero is negative infinity. In Table A.10, under *TP*1, we indicate with *** when the average is taken over the 33 finite terms, ** with an average over 34 terms and * over 35 terms. Plotting leads to Chart 6.2.

Chart 6.2: Average extra digits between single and double precision for all procedures and test problems

(See Chart B.10: Average extra digits between Single and Double Precision calculations)

Observe from Table A1.10 that Procedure 9 gives the largest average number of extra digits provided by the double precision over the single precision calculations for all the test problems.

Heyman and O'Leary [8] used two algorithms for computing *M,* without previously computing *Z* or $A^{\#}$ – an algorithm *MH,* which is based on the two stage *UL* factorisation, using U^{-1} and *L* factors as in Heyman's method (similar to our Procedure 12), and an algorithm *M,* using the *UL* factors and normalisation. They obtained values, for the number of extra accurate digits, between 6 and 7 for all *TPs* for their algorithm *M* but

displayed widely varying values for different *TPs* for the algorithm *M*. They also showed that if Z or $A^{\#}$ were known in advance then the *M* matrix could only be computed accurately from *Z* or from $A^{\#}$ (as special cases of our (2.4)) for *TP*1, *TP*3, and *TP*41.

Heyman and Reeves [9] presented four algorithms - *LINPACK, SR, KSGTH*, and *KSGAUSS* for computing *M* with the same test problems used in this paper. They explored different software packages deciding that *LINPACK* "worked the best". The *KSGTH* is the same as our Procedure 1 using the *GTH* algorithm to compute the steady state probabilities while *KSGAUSS* used Gaussian elimination to solve the stationary equations. Their favoured algorithm, is the *SR*, "State reduction", procedure of Kohlas on which Procedure 9 is based. The Hunter *EGTH* Procedure consistently produces results in the range 7.30 to 7.43, similar to that achieved by Heyman & Reeves [9] (as extrapolated from their graphical output), which was not exceeded by any other procedure, either in [9] or this paper.

The solution computed by LINPACK did not run for *TP*44 as the matrix inverse could not be computed. In our calculations the only test problem and procedure that could not be computed accurately was for *TP*44 under Procedure 4 (updating the Group inverse under Pert AL2 using single precision). *MATLAB* enabled us to carry out all of the other calculations. An interesting observation for this particular ill-conditioned test problem is that Procedure 3 (updating a G-inverse under Pert AL1) gives the most accurate overall residual errors.

There is a range of other comparisons that we can make, using the results given in Appendix 1, but we generally end up pointing to Hunter's extended *GTH* Procedure as giving the most accurate results. Our general recommendation is to use this procedure where possible.

When paper [18] was written it was hoped that the perturbation procedures were going to generally yield accurate results, comparable with other procedures. Apart from isolated situations, Procedures 3 – 8 do not perform as well as we had hoped. The *EGTH* procedure, involving no subtractions generally overshadows the perturbation procedures. Procedure 12, while not in the same class as Procedure 9, reliably produces the second most accurate results.

Appendix A: Error calculations for all Procedures and all Test Problems

Table A.1: MINRE under Single Precision

MINRE(S)	TP1	TP2	TP3	TP41	TP42	TP43	TP44
Procedure 1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1.0902E-02
Procedure 2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Procedure 3	0.0000E+00	2.8133E-05	1.5333E-02	1.1921E-07	3.3379E-06	1.9091E-04	1.9091E-04
Procedure 4	0.0000E+00	2.8014E-06	1.2846E-03	0.0000E+00	2.5034E-06	1.4853E-04	NaN
Procedure 5	0.0000E+00	9.9784E-06	1.2996E-03	0.0000E+00	1.1921E-06	1.8179E-04	4.9167E-02
Procedure 6	0.0000E+00	1.0908E-05	1.2996E-03	0.0000E+00	5.8413E-06	8.6933E-04	1.0224E-01
Procedure 7	0.0000E+00	5.7817E-06	1.2996E-03	0.0000E+00	1.5497E-06	3.1388E-04	4.2600E-02
Procedure 8	0.0000E+00	1.7941E-05	1.2996E-03	0.0000E+00	4.4107E-06	2.2805E-04	7.9652E-02
Procedure 9	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Procedure 10	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8.6164E-03
Procedure 11	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8.6164E-03
Procedure 12	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00

Table A.2: MINRE under Double Precision

MINRE(D)	TP1	TP2	TP3	TP41	TP42	TP43	TP44
Procedure 1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Procedure 2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Procedure 3	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Procedure 4	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Procedure 5	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Procedure 6	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Procedure 7	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Procedure 8	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Procedure 9	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Procedure 10	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Procedure 11	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Procedure 12	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00

Table A.3: MAXRE under Single Precision

MAXRE(S)	TP1	TP2	TP3	TP41	TP42	TP43	TP44
Procedure 1	1.5259E-05	2.3131E-03	1.0000E+00	7.6294E-06	7.3242E-04	9.3750E-02	1.9313E+01
Procedure 2	3.0518E-05	1.9531E-03	1.0000E+00	1.5259E-05	9.7656E-04	6.2500E-02	7.7565E+00
Procedure 3	5.8830E-05	1.9503E-03	7.4928E-01	9.0599E-06	8.2135E-04	5.8108E-02	5.8108E-02
Procedure 4	3.0518E-05	1.3188E-03	7.6065E-01	6.4373E-06	7.2205E-04	4.4751E-02	NaN
Procedure 5	5.7161E-05	1.1806E-03	1.4523E+00	5.8413E-06	3.8600E-04	5.0105E-02	2.7472E+01
Procedure 6	2.2531E-05	1.5189E-03	1.4884E+00	5.7220E-06	3.3975E-04	4.0965E-02	1.1226E+01
Procedure 7	8.1241E-05	1.9780E-03	1.4482E+00	5.0068E-06	3.8671E-04	4.6234E-02	2.5369E+01
Procedure 8	2.4557E-05	1.5258E-03	1.4909E+00	4.8876E-06	3.6669E-04	4.6220E-02	1.1133E+01
Procedure 9	6.1035E-05	1.9531E-03	5.3333E-01	7.6294E-06	4.9897E-04	8.5969E-02	5.0000E+00
Procedure 10	6.1035E-05	1.9531E-03	1.6523E+04	1.5259E-05	9.7656E-04	7.5120E-02	7.0000E+00
Procedure 11	6.1035E-05	1.9531E-03	1.6522E+04	7.6294E-06	9.7656E-04	7.5120E-02	7.0000E+00
Procedure 12	1.5259E-05	1.9531E-03	1.6522E+04	7.6294E-06	4.8828E-04	7.1746E-02	9.0000E+00

Table A.4: MAXRE under Double Precision

MAXRE(D)	TP1	TP2	TP3	TP41	TP42	TP43	TP44
Procedure 1	1.1369E-13	2.4714E-12	1.8626E-09	1.4211E-14	1.8190E-12	1.1642E-10	1.4901E-08
Procedure 2	5.6843E-14	3.6380E-12	1.4461E-09	1.4211E-14	9.0949E-13	1.7462E-10	7.4506E-09
Procedure 3	2.2737E-13	5.4570E-12	2.5251E-09	6.1950E-14	6.3665E-12	3.4925E-10	3.4925E-10
Procedure 4	1.1369E-13	2.3647E-11	4.2285E-05	2.1316E-14	2.3099E-12	1.7462E-10	2.2352E-08
Procedure 5	2.8422E-13	4.7893E-12	1.9447E-09	1.5987E-14	2.3055E-12	2.3283E-10	2.9802E-08
Procedure 6	1.1369E-13	3.6380E-12	1.8626E-09	1.4211E-14	9.0949E-13	1.1642E-10	7.5181E-09
Procedure 7	1.1369E-13	3.6380E-12	1.8626E-09	1.4211E-14	1.8190E-12	1.7462E-10	1.4901E-08
Procedure 8	1.1369E-13	4.2912E-12	1.8626E-09	2.8422E-14	3.6380E-12	3.4925E-10	2.0740E-08
Procedure 9	1.1369E-13	3.6380E-12	1.4461E-09	1.4211E-14	1.8190E-12	1.1642E-10	7.4506E-09
Procedure 10	5.6843E-14	3.7313E-12	2.0940E-05	1.4211E-14	1.8190E-12	1.7462E-10	2.2352E-08
Procedure 11	5.6843E-14	3.7313E-12	2.0941E-05	1.4211E-14	1.8190E-12	1.7462E-10	2.2352E-08
Procedure 12	1.1369E-13	3.7313E-12	2.0941E-05	1.4211E-14	1.8190E-12	1.1642E-10	1.4901E-08

Table A.5: ORE under Single Precision

ORE(S)	TP1	TP2	TP3	TP41	TP42	TP43	TP44
Procedure 1	7.9721E-05	2.8165E-02	3.7670E+00	2.1267E-04	1.7017E-02	1.8252E+00	5.2130E+02
Procedure 2	8.4311E-05	1.9942E-02	5.1353E+00	3.4666E-04	1.4591E-02	1.4978E+00	1.4851E+02
Procedure 3	2.6092E-04	3.5845E-02	5.4519E+00	2.6882E-04	2.8050E-02	1.6858E+00	1.6858E+00
Procedure 4	1.3366E-04	1.8962E-02	3.6441E+00	1.7709E-04	1.5828E-02	1.4867E+00	NaN
Procedure 5	1.7825E-04	1.8865E-02	5.8119E+00	1.4186E-04	9.3399E-03	1.1775E+00	3.3758E+02
Procedure 6	1.2526E-04	1.9844E-02	6.0251E+00	1.2279E-04	9.7828E-03	8.9603E-01	1.9242E+02
Procedure 7	2.4018E-04	1.9384E-02	5.7981E+00	1.3071E-04	1.0204E-02	9.7736E-01	2.3153E+02
Procedure 8	1.2872E-04	1.6985E-02	6.0274E+00	1.2934E-04	9.7471E-03	9.3048E-01	1.9473E+02
Procedure 9	2.0275E-04	1.6383E-02	3.5601E+00	1.1259E-04	5.3112E-03	7.8108E-01	8.5883E+01
Procedure 10	1.8367E-04	2.7213E-02	3.3883E+04	2.2340E-04	1.7498E-02	2.1203E+00	1.7003E+02
Procedure 11	1.7092E-04	2.7842E-02	3.3883E+04	2.4676E-04	1.7498E-02	2.1203E+00	1.7203E+02
Procedure 12	7.1913E-05	2.6956E-02	3.3883E+04	1.9002E-04	1.4920E-02	1.6139E+00	1.5876E+02

Table A.6: ORE under Double Precision

ORE(D)	TP1	TP2	TP3	TP41	TP42	TP43	TP44
Procedure 1	3.1486E-13	4.2940E-11	8.2400E-09	4.3465E-13	4.1593E-11	3.4351E-09	3.2901E-07
Procedure 2	2.9809E-13	3.8647E-11	7.2145E-09	4.4076E-13	2.8369E-11	4.0932E-09	1.9688E-07
Procedure 3	1.1076E-12	1.0909E-10	1.6599E-08	1.6265E-12	1.0069E-10	8.7043E-09	8.7043E-09
Procedure 4	4.0834E-13	3.5271E-10	1.6354E-04	6.8556E-13	6.8057E-11	4.7508E-09	5.2569E-07
Procedure 5	5.8542E-13	8.0202E-11	1.4783E-08	7.6406E-13	6.9908E-11	6.9648E-09	7.2536E-07
Procedure 6	3.1319E-13	3.9636E-11	6.9062E-09	3.8036E-13	2.9394E-11	3.2306E-09	2.2832E-07
Procedure 7	3.6315E-13	4.8937E-11	1.1378E-08	3.6848E-13	4.4544E-11	4.6604E-09	4.2368E-07
Procedure 8	4.5819E-13	1.0442E-10	1.5714E-08	8.0280E-13	8.5687E-11	1.1774E-08	6.2413E-07
Procedure 9	2.9554E-13	2.8481E-11	5.2755E-09	2.8832E-13	1.9558E-11	1.5769E-09	1.4170E-07
Procedure 10	2.4492E-13	4.1265E-11	5.2111E-05	3.4472E-13	3.7945E-11	3.1286E-09	3.4516E-07
Procedure 11	2.1694E-13	4.2175E-11	5.2112E-05	3.2230E-13	3.7490E-11	3.3033E-09	3.4516E-07
Procedure 12	2.8488E-13	3.6427E-11	5.2112E-05	2.9698E-13	3.0710E-11	3.0476E-09	2.5663E-07

Table A.7 for MINE(S, D) between Single and Double Precision calculations

MIN(S, D)	TP1	TP2	TP3	TP41	TP42	TP43	TP44
Procedure 1	2.8422E-14	1.8878E-08	8.9733E-08	3.7478E-08	2.5415E-07	2.5415E-07	2.5415E-07
Procedure 2	4.4409E-16	3.7599E-05	2.9789E-02	1.0393E-07	8.1511E-09	3.4234E-07	2.1804E-07
Procedure 3	4.6971E-08	3.8719E-05	1.4642E-02	4.9425E-08	1.1721E-04	1.4576E-03	1.4576E-03
Procedure 4	1.0398E-07	4.2477E-05	2.1705E-02	1.3248E-07	2.8595E-05	4.7612E-04	Inf
Procedure 5	9.5415E-09	3.0867E-04	3.0409E-03	2.5835E-08	3.8400E-06	4.0875E-04	7.5673E-02
Procedure 6	3.8137E-09	1.8545E-04	3.0409E-03	4.4225E-09	2.8390E-06	2.8088E-04	5.6237E-02
Procedure 7	3.0546E-08	3.3281E-04	3.0409E-03	3.8417E-07	5.2825E-05	5.2960E-03	6.7380E-01
Procedure 8	1.2641E-09	1.8556E-04	3.0409E-03	1.6998E-08	2.7935E-06	2.8083E-04	5.6237E-02
Procedure 9	0.0000E+00	7.8481E-08	1.1097E-07	1.0259E-08	1.5811E-08	2.3757E-08	5.6944E-09
Procedure 10	0.0000E+00	1.8878E-08	8.9733E-08	1.2903E-07	9.2766E-08	2.5415E-07	2.5415E-07
Procedure 11	0.0000E+00	1.8878E-08	8.9733E-08	1.2903E-07	9.2766E-08	2.5415E-07	2.5415E-07
Procedure 12	0.0000E+00	1.8878E-08	8.9733E-08	3.7478E-08	3.7478E-08	3.7478E-08	3.7478E-08

Table A.8: MAXE(S, D) errors between Single and Double Precision calculations

MAX(S, D)	TP1	TP2	TP3	TP41	TP42	TP43	TP44
Procedure 1	2.7127E-05	4.5160E-01	1.3403E+05	5.8003E-05	6.8901E-02	6.7987E+02	1.0528E+08
Procedure 2	4.9506E-05	3.7231E+00	9.7505E+04	5.0969E-05	3.8604E-01	3.6333E+03	2.5358E+07
Procedure 3	6.9505E-03	3.2695E+00	5.5296E+05	1.4576E-04	1.0315E+00	7.0150E+03	7.0150E+03
Procedure 4	2.9093E-03	1.7307E+00	2.6785E+05	1.0287E-04	2.3300E+00	1.2735E+04	1.0000E+10
Procedure 5	2.8536E-03	1.8751E+00	1.8137E+05	5.3411E-05	3.4919E-01	3.6567E+03	1.2861E+08
Procedure 6	2.5588E-03	8.9710E-01	1.8137E+05	3.8305E-05	2.4802E-01	2.6363E+03	4.9001E+07
Procedure 7	2.2191E-03	2.0862E+00	1.8137E+05	5.7456E-05	3.4977E-01	3.6567E+03	1.2861E+08
Procedure 8	2.5609E-03	8.9731E-01	1.8137E+05	3.9159E-05	2.4806E-01	2.6363E+03	4.9001E+07
Procedure 9	6.7817E-05	2.2032E-03	1.2214E+00	1.4700E-05	1.0511E-03	1.1536E-01	7.6999E+00
Procedure 10	8.8162E-05	1.4282E+00	1.8206E+05	1.9367E-05	6.0751E-02	6.1944E+02	7.3597E+06
Procedure 11	8.8162E-05	1.4282E+00	1.8206E+05	2.0312E-05	6.0751E-02	6.1944E+02	7.3597E+06
Procedure 12	2.7127E-05	1.4272E+00	1.8206E+05	1.7415E-05	6.0359E-02	6.1942E+02	7.3597E+06

Table A.9: REL(S, D) errors between Single and Double Precision calculations

REL(S, D)	TP1	TP2	TP3	TP41	TP42	TP43	TP44
Procedure 1	6.2602E-05	1.8238E-04	9.7760E+00	8.5381E+05	2.0605E-03	2.9959E+00	2.9939E+04
Procedure 2	4.2048E-05	2.2072E-04	5.5973E+01	6.8745E+05	9.9624E-04	9.6408E+00	9.0833E+04
Procedure 3	8.0903E-05	4.1639E-02	5.6868E+01	3.7384E+06	4.9278E-03	3.4096E+01	2.2813E+05
Procedure 4	1.1128E-04	1.7467E-02	1.8265E+01	1.5096E+06	2.2607E-03	3.1956E+01	2.1517E+05
Procedure 5	2.5273E-05	1.7052E-02	2.9542E+01	1.5219E+06	1.7384E-03	1.2950E+01	1.3674E+05
Procedure 6	2.3256E-05	1.5258E-02	1.8216E+01	1.5219E+06	1.4254E-03	1.0783E+01	1.1486E+05
Procedure 7	4.2810E-05	1.3236E-02	3.3603E+01	1.5219E+06	1.5823E-03	1.0819E+01	1.1517E+05
Procedure 8	2.4030E-05	1.5273E-02	1.8219E+01	1.5219E+06	1.4598E-03	1.0785E+01	1.1486E+05
Procedure 9	2.5894E-05	2.9606E-04	2.6209E-02	5.3162E+00	2.4325E-04	1.3152E-02	1.1988E+00
Procedure 10	1.0148E-04	4.8392E-04	4.0830E+01	1.5125E+06	4.9828E-04	2.6174E+00	2.7278E+04
Procedure 11	9.1177E-05	4.9096E-04	4.0828E+01	1.5125E+06	4.7100E-04	2.6174E+00	2.7278E+04
Procedure 12	1.0511E-04	1.6090E-04	4.0822E+01	1.5125E+06	4.4613E-04	2.6199E+00	2.7278E+04

Table A.10: Average extra digits between Single and Double Precision calculations

Average extra digits	TP1	TP2	TP3	TP41	TP42	TP43	TP44
Procedure 1	***6.7522	4.9691	3.2245	6.5086	5.0423	3.2272	0.7819
Procedure 2	**7.1182	4.5133	2.4592	6.7433	5.0996	3.5632	2.3108
Procedure 3	6.2773	3.9600	1.7416	6.0373	3.9882	2.1693	2.1693
Procedure 4	6.3811	4.5148	1.8947	6.3439	4.0975	2.1999	NaN
Procedure 5	6.8213	4.3463	2.1006	6.5652	4.5249	2.5096	0.1074
Procedure 6	6.7783	4.3526	2.1006	6.7684	4.8025	2.7873	0.5022
Procedure 7	6.8009	4.1487	2.1006	6.4926	4.4743	2.4530	0.1447
Procedure 8	6.9079	4.3525	2.1006	6.7396	4.8026	2.7873	0.5022
Procedure 9	*7.3504	7.2928	7.3526	7.3681	7.4157	7.4296	7.3321
Procedure 10	*6.6272	4.5093	2.9679	6.8748	5.1722	3.3115	1.4794
Procedure 11	*6.5949	4.5091	2.9679	6.9339	5.1722	3.3115	1.4794
Procedure 12	***6.7660	4.5436	2.9679	7.0429	5.5830	4.1590	2.7397

Appendix B: Charts of errors for all Procedures and all Test Problems

Chart B.1: MINRE under Single Precision

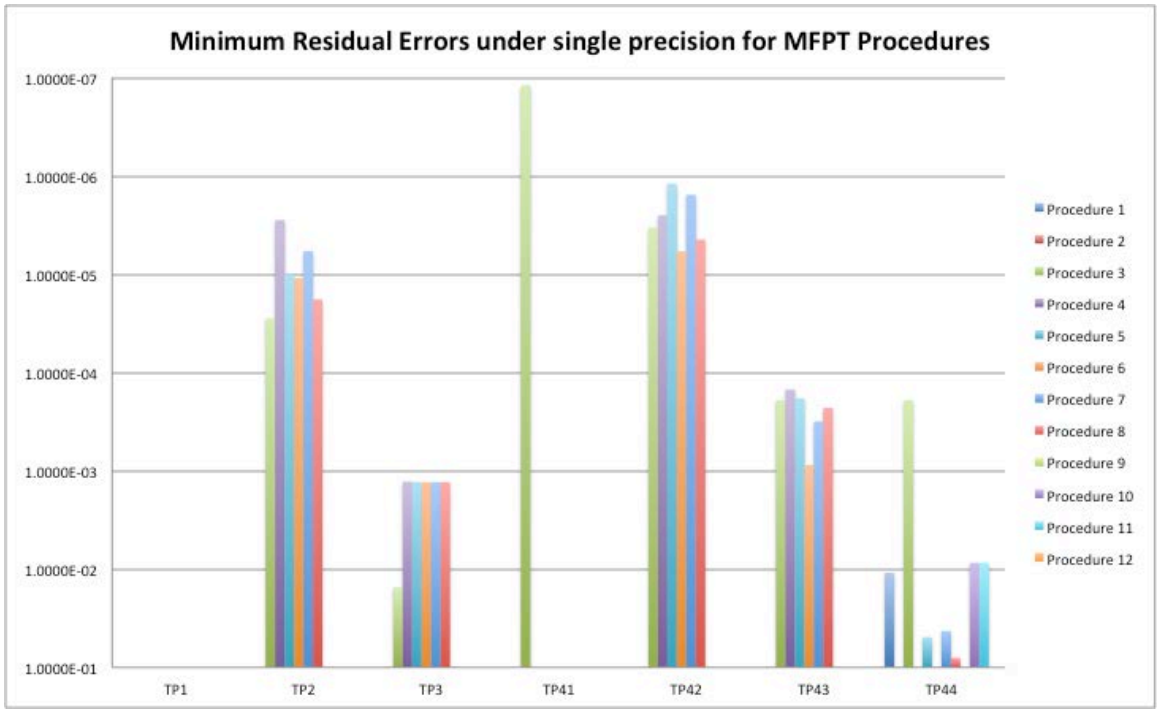


Chart B.2: MINRE under Double Precision

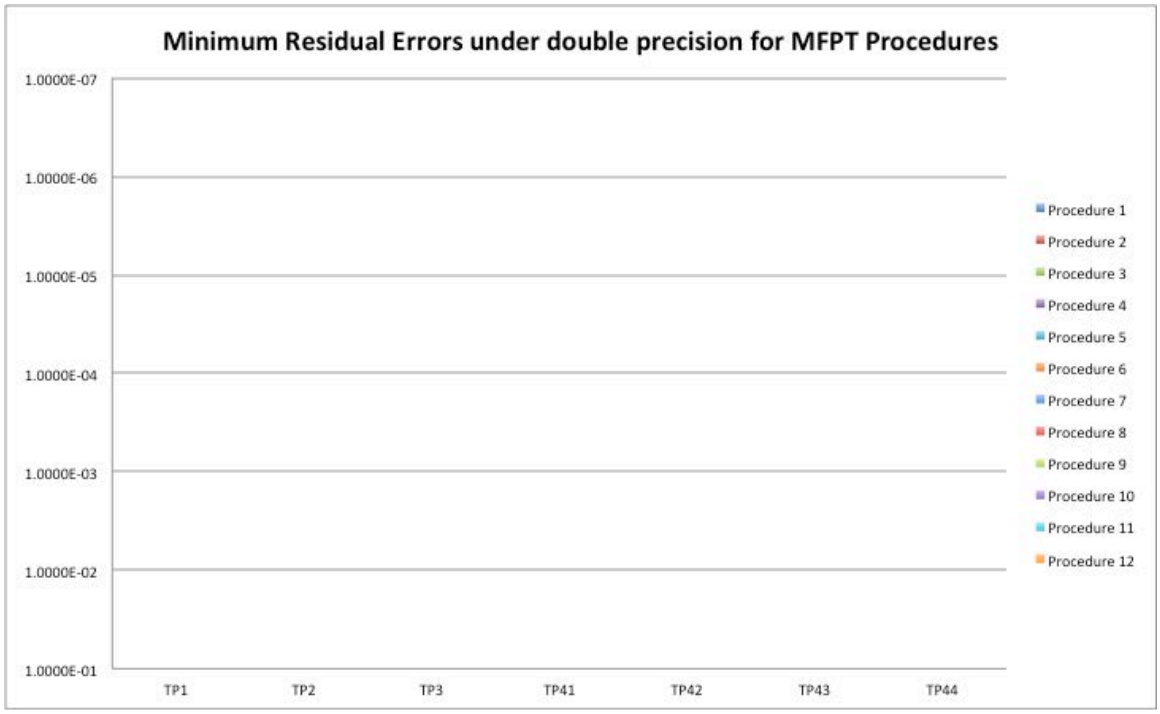


Chart B.3: MAXRE under Single Precision

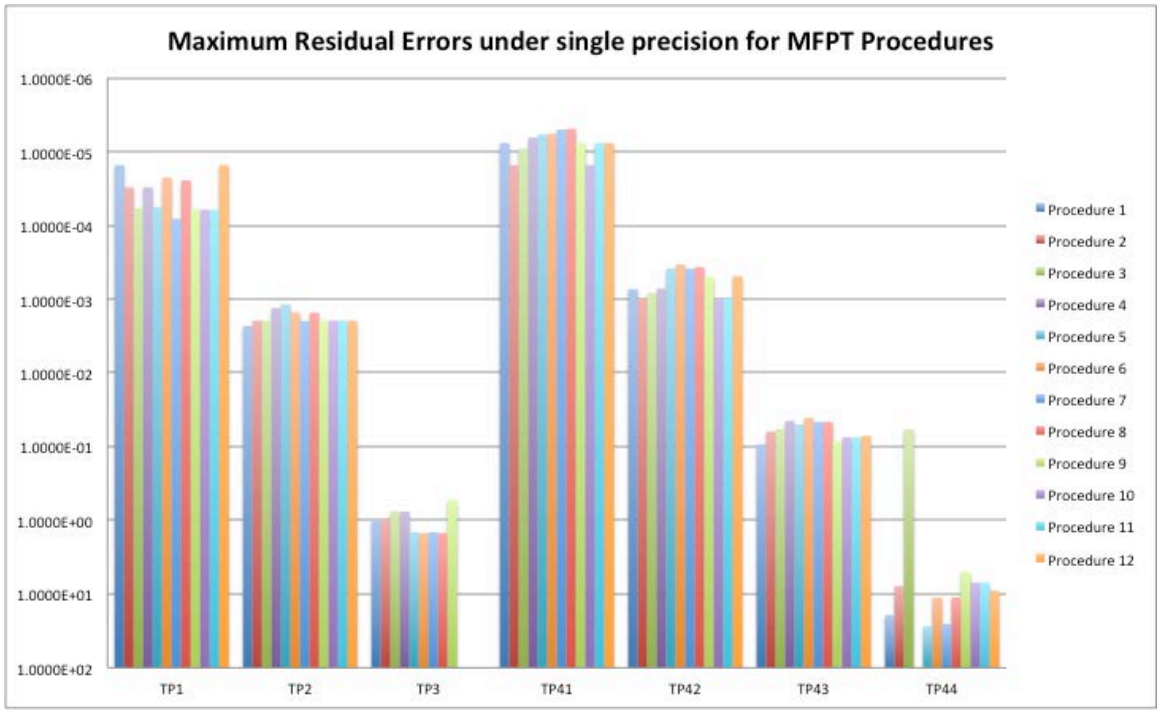


Chart B.4: MAXRE under Double Precision

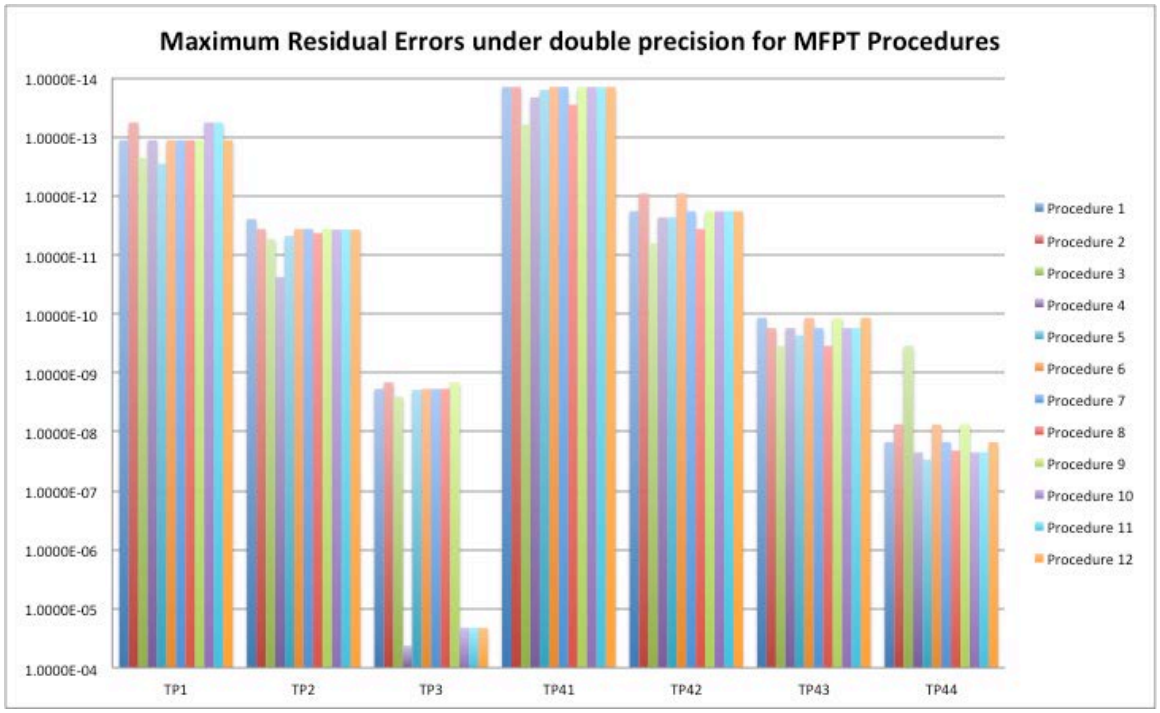


Chart B.5: ORE under Single Precision

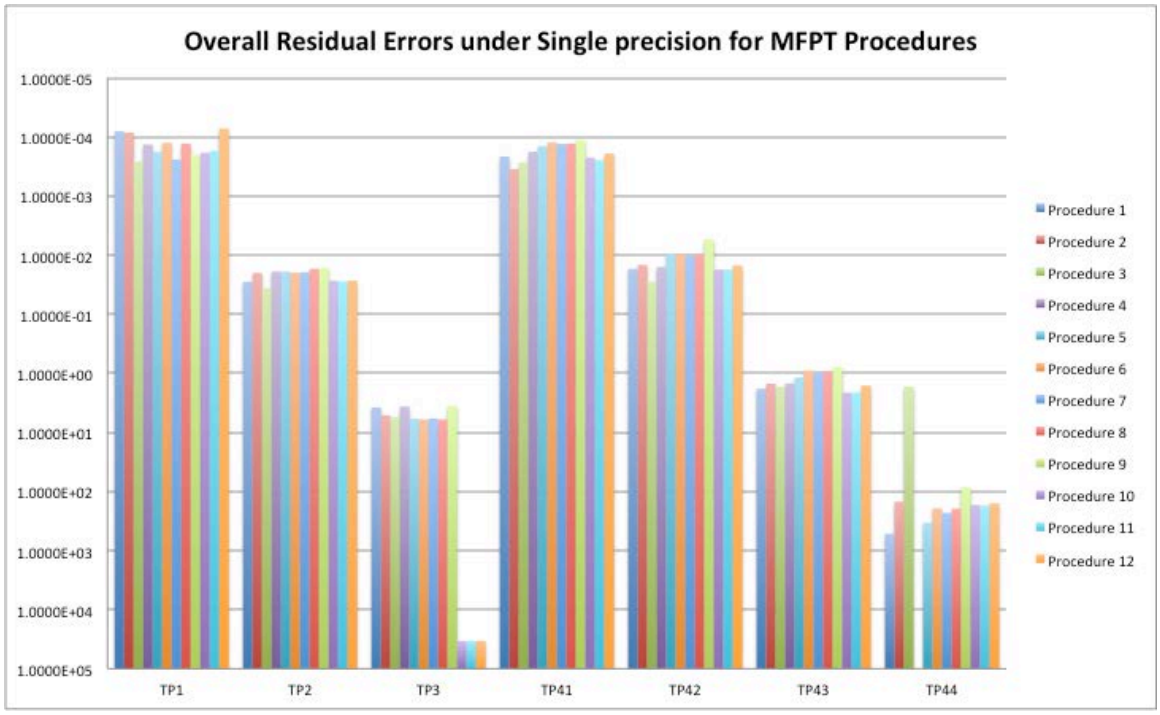


Chart B.6: ORE under Double Precision

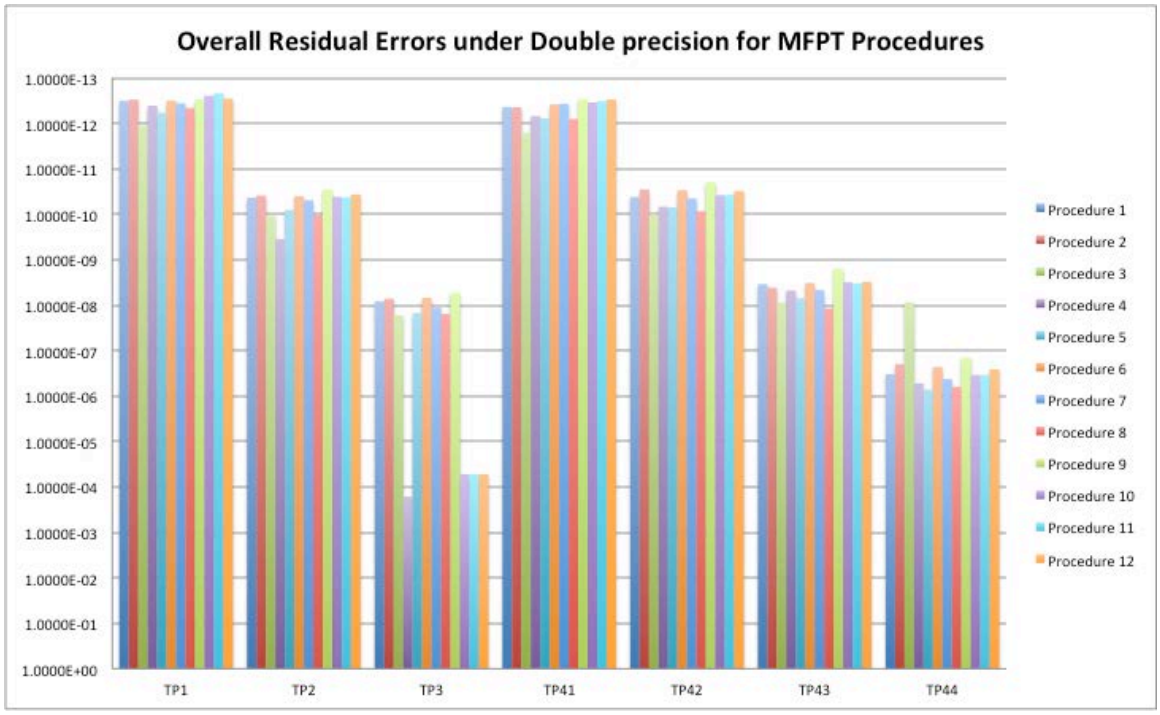


Chart B.7 for MINE(S, D) between Single and Double Precision calculations

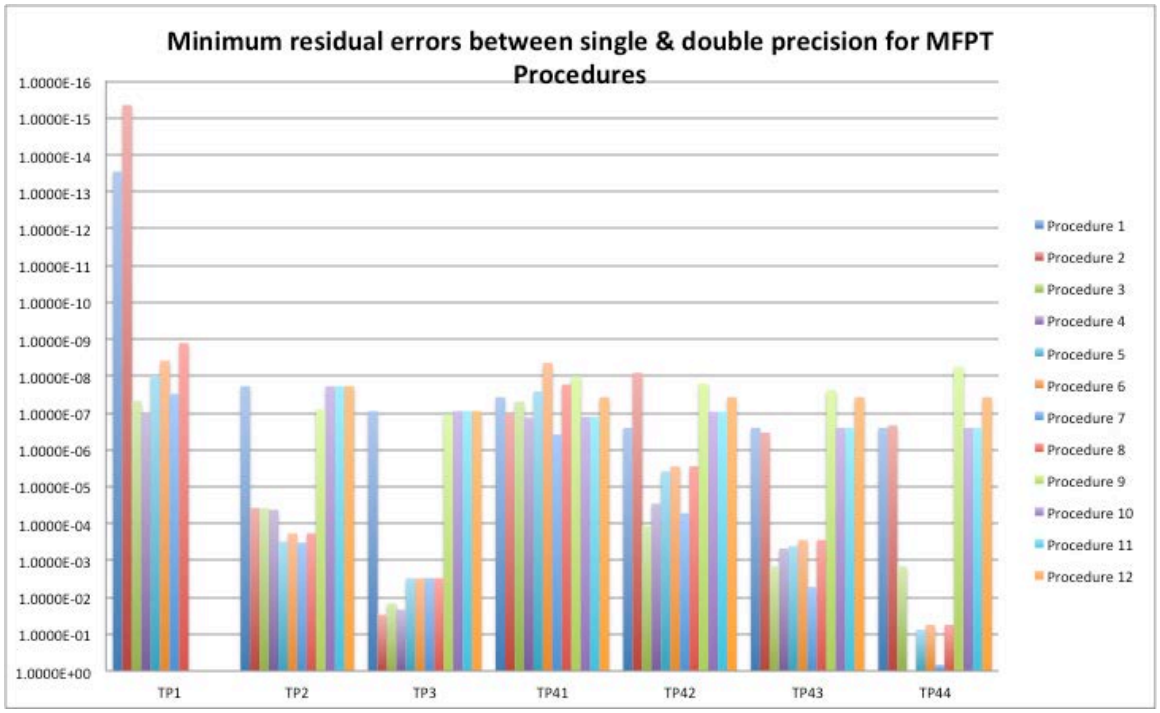


Chart B.8: MAXE(S, D)) errors between Single and Double Precision calculations

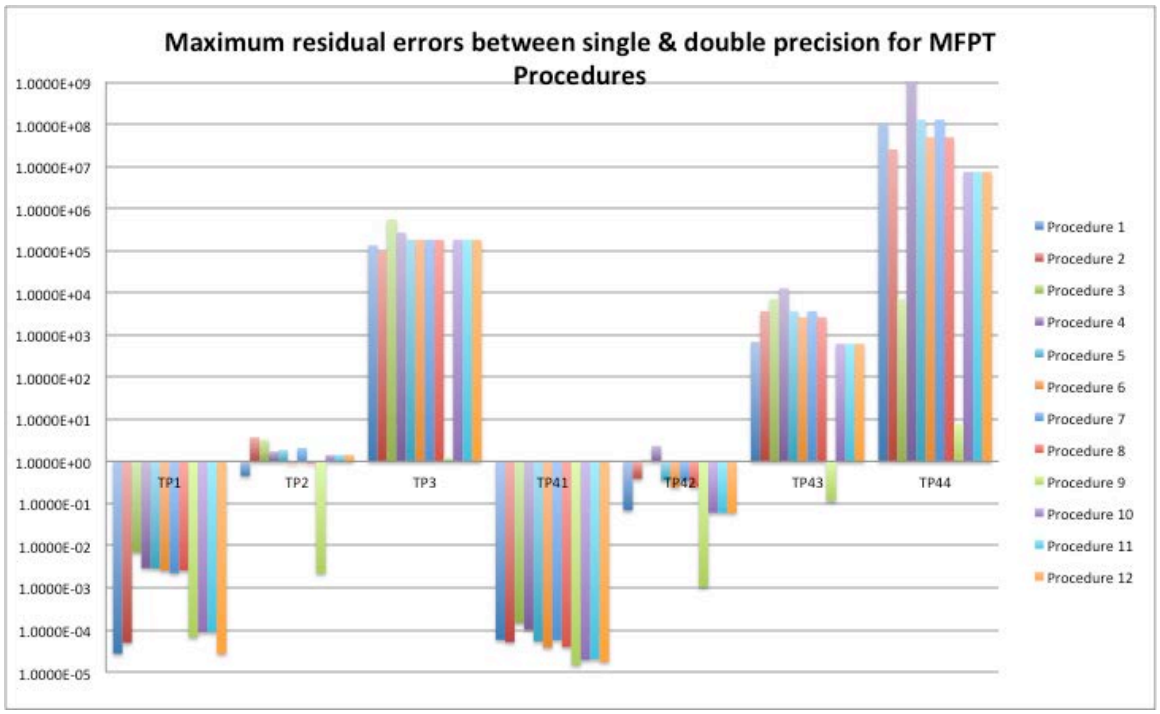
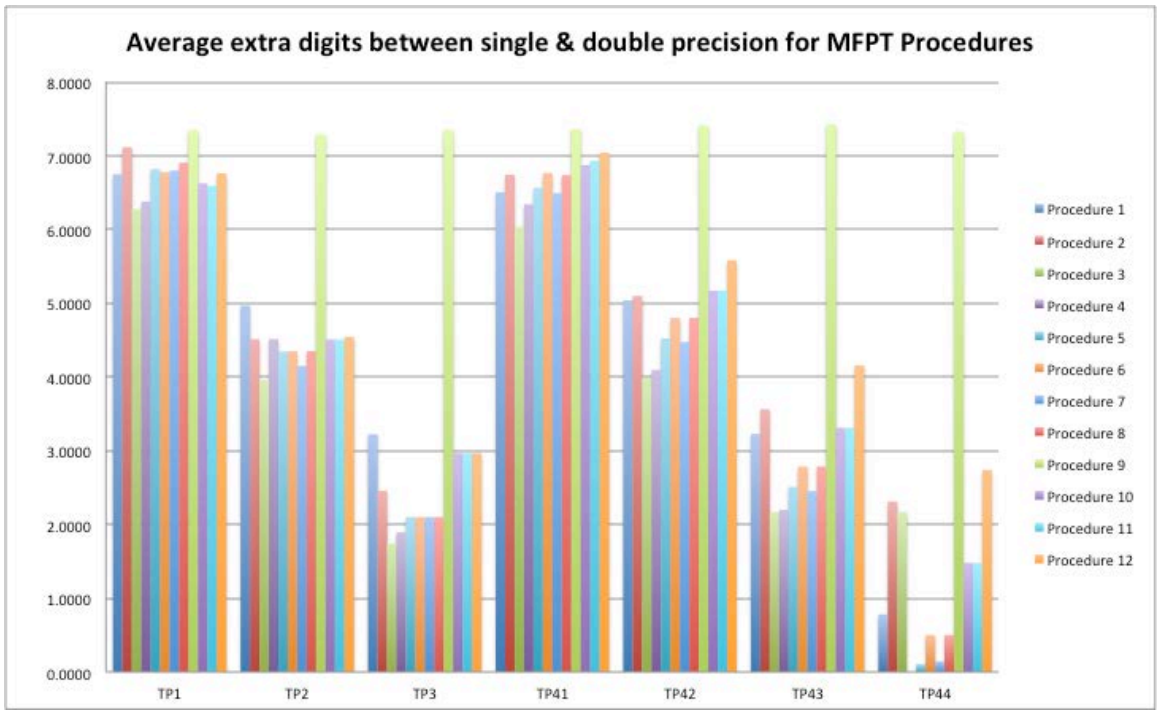


Chart B.9: REL(S, D) errors between Single and Double Precision calculations

Chart B.10: Average extra digits between Single and Double Precision calculations



References

[1] Benzi, M. A direct projection method for Markov chains. Linear Algebra Appl, **386**, (2004), 27-49.

[2] D.A. Bini, G. Latouche, B. Meini. *Numerical Methods for Structured Markov Chains*, Oxford University Press, New York. (2005).

[3] W. K. Grassman, M.I. Taksar, D.P. Heyman, Regenerative analysis and steady state distributions for Markov chains. Oper. Res. **33** (1985) 1107-1116.

[4] W. K. Grassman, Means and variances in Markov reward systems, in *Linear Algebra, Markov Chains and Queuing Models*, C.D. Meyer and R.J. Plemmons. Eds, Springer-Verlag, New York, (1993), 193-204.

[5] W.J. Harrod, R.J. Plemmons, Comparison of some direct methods for computing stationary distributions of Markov chains. SIAM J Sci Stat Comput, **5** (1984), 463-479.

[6] D.P. Heyman, Further comparisons of direct methods for computing stationary distributions of Markov chains. SIAM J. Alg. Disc Meth. **8**, (1987) 226-232.

[7] D.P. Heyman, Accurate computation of the Fundamental Matrix of a Markov chain, SIAM J. Matrix Anal. Appl. **16** (1995) 954-963.

[8] Heyman, D.P., O'Leary, D.P. What is fundamental for Markov chains: First Passage Times, Fundamental matrices, and Group Generalized Inverses, *Computations with Markov Chains,* Chap 10, 151-161, Ed W.J. Stewart, Springer. New York, (1995).

[9] D.P. Heyman, A. Reeves, Numerical solutions of linear equations arising in Markov chain models. ORSA J. Comp. **1** (1989) 52-60.

[10] J.J. Hunter, On the moments of Markov renewal processes. Adv. in Appl. Probab. **1** (1969) 188-210.

[11] J.J. Hunter, Generalized inverses and their application to applied probability problems. Linear Algebra Appl. **45** (1982) 157-198.

[12] J.J. Hunter, *Mathematical Techniques of Applied Probability, Volume 2, Discrete Time Models: Techniques and Applications*, Academic, New York. (1983).

[13] J.J. Hunter, Characterisations of generalized inverses associated with Markovian kernels, Linear Algebra Appl. **102** (1988) 121-142.

[14] J.J. Hunter, The computation of stationary distributions of Markov chains through perturbations. J Appl Math Stoch. Anal. **4** (1991) 29-46.

[15] J.J. Hunter, Simple procedure for finding mean first passage times in Markov chains. Asia Pac. J. Oper. *Res*. **24** (2007) 813-829.

[16] J. J. Hunter, Generalized inverses of Markovian kernels in terms of properties of the Markov chain. Linear Algebra Appl. **447** (2014) 38-55.

[17] J.J. Hunter, Accurate calculations of stationary distributions and mean first passage times in Markov renewal processes and Markov chains. Special Matrices **4** (2016) 151-175.

[18] J.J. Hunter, The computation of the key properties of Markov Chains via Perturbations, Linear Algebra Appl. **511** (2016) 176-202.

[19] J. G. Kemeny, J.L. Snell, *Finite Markov chains*, Von Nostrand, New York, 1960.

[20] J. Kohlas, Numerical computation of mean passage times and absorption probabilities in Markov and semi-Markov models. Z. Oper.- Res. **30** (1986) 197-207.

[21] C.D. Meyer Jr., The role of the group generalized inverse in the theory of Markov chains. SIAM Rev. **17** (1975) 443-464.

[22] J. Sherman, W.J. Morrison, Adjustment of an Inverse Matrix Corresponding to Changes in the Elements of a Given Column or a Given Row of the Original Matrix (abstract). Ann Math Stat **20** (1949) 621.

[23] T.J. Sheskin, A Markov partitioning algorithm for computing steady state probabilities. Oper. Res. **33** (1985) 228-235.